

501P07730800

日 本 国 特 許 庁
JAPAN PATENT OFFICE

5
9/27



別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2000年 5月22日

出 願 番 号

Application Number:

特願2000-154424

出 願 人

Applicant(s):

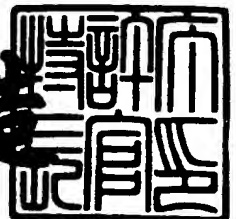
ソニー株式会社



2001年 4月20日

特許庁長官
Commissioner,
Japan Patent Office

及 川 耕 造



出証番号 出証特2001-3031702

【書類名】 特許願

【整理番号】 0000029802

【提出日】 平成12年 5月22日

【あて先】 特許庁長官殿

【国際特許分類】 G11C 7/00

【発明者】

 【住所又は居所】 東京都品川区北品川6丁目7番35号 ソニー株式会社
 内

 【氏名】 山崎 友敬

【発明者】

 【住所又は居所】 東京都品川区北品川6丁目7番35号 ソニー株式会社
 内

 【氏名】 馬場 博光

【特許出願人】

 【識別番号】 000002185

 【氏名又は名称】 ソニー株式会社

 【代表者】 出井 伸之

【代理人】

 【識別番号】 100086841

 【弁理士】

 【氏名又は名称】 脇 篤夫

【手数料の表示】

 【予納台帳番号】 014650

 【納付金額】 21,000円

【提出物件の目録】

 【物件名】 明細書 1

 【物件名】 図面 1

 【物件名】 要約書 1

 【包括委任状番号】 9710074

特 2 0 0 0 - 1 5 4 4 2 4

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 情報処理装置、情報処理方法

【特許請求の範囲】

【請求項 1】 アプリケーションソフトウェア又はデータファイルが記憶可能な内部記憶手段と、

アプリケーションソフトウェア又はデータファイルが記憶可能な外部記憶部との間で情報の入出力を可能とする接続手段と、

上記接続手段により接続された上記外部記憶部に記憶されているアプリケーションソフトウェア又はデータファイルを認識する認識手段と、

上記内部記憶手段に記憶されたアプリケーションソフトウェア又はデータファイル、及び上記認識手段で認識されたアプリケーションソフトウェア又はデータファイルを提示すると共に、提示されたアプリケーションソフトウェアの起動又はデータファイルの使用を選択することができるようになされたユーザーインターフェース手段と、

上記ユーザーインターフェース手段による操作により、上記外部記憶部に記憶されているアプリケーションソフトウェアの起動が求められた際に、そのアプリケーションソフトウェアを上記内部記憶手段にロードした上で起動処理を行う起動処理制御手段と、

を備えたことを特徴とする情報処理装置。

【請求項 2】 上記ユーザーインターフェース手段による操作により、もしくは起動中のアプリケーションソフトウェアの処理により、上記外部記憶部に記憶されているデータファイルの参照が求められた際に、そのデータファイルを上記内部記憶手段にロードした上で、参照処理が行われるようにする参照処理制御手段を、さらに備えたことを特徴とする請求項 1 に記載の情報処理装置。

【請求項 3】 上記外部記憶部は可搬性記録媒体であり、上記接続手段は、可搬性記録媒体に対する記録再生部として実現されることを特徴とする請求項 1 に記載の情報処理装置。

【請求項 4】 上記外部記憶部は有線又は無線の通信路を介して接続される外部サーバであり、上記接続手段は、上記通信路を介して通信を行う通信部とし

て実現されることを特徴とする請求項 1 に記載の情報処理装置。

【請求項 5】 アプリケーションソフトウェアの起動操作を検知する起動検知手順と、

上記起動検知手順で起動操作を検知したアプリケーションソフトウェアが内部記憶手段に記憶されているか、或いは接続された外部記憶部に記憶されているかを判別する判別手順と、

上記判別手順において、上記外部記憶部に記憶されていると判別された場合は、上記外部記憶部からアプリケーションソフトウェアを上記内部記憶手段にロードするロード手順と、

上記ロード手順により上記内部記憶手段にロードされたアプリケーションソフトウェアを起動する起動手順と、

が行われることを特徴とする情報処理方法。

【請求項 6】 ユーザーインターフェースによる操作により、もしくは起動中のアプリケーションソフトウェアの処理により、上記外部記憶部に記憶されているデータファイルの参照が求められた際に、そのデータファイルを上記内部記憶手段にロードした上で、参照処理が行われるようにするデータファイル参照手順が行われることを特徴とする請求項 5 に記載の情報処理方法。

【請求項 7】 上記外部記憶部は可搬性記録媒体であり、上記ロード手順では、可搬性記録媒体に対する再生動作を行ってロードを実行することを特徴とする請求項 5 に記載の情報処理方法。

【請求項 8】 上記外部記憶部は有線又は無線の通信路を介して接続される外部サーバであり、上記ロード手順は、上記通信路を介して通信を行ってロードを実行することを特徴とする請求項 5 に記載の情報処理方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、情報処理装置及び情報処理方法に係り、特に内部記憶手段を有すると共に外部記憶部から情報入力を行うことができる場合に関するものである。

【 0 0 0 2 】

【従来の技術】

パーソナルコンピュータやPDA (Personal Digital Assistants: 携帯情報機器) などの情報処理装置においては、装置内部に設けられる内部記憶部としてはRAM、フラッシュメモリ等の固体メモリやHDD (Hard Disc Drive) などがあり、また装置外部の記憶部として、光ディスク、光磁気ディスク、磁気ディスク、メモリカードなどの各種の可搬性の記録媒体を利用できるようにされている。

【 0 0 0 3 】

【発明が解決しようとする課題】

ところでPDA機器などでは、或るアプリケーションソフトウェアを起動したり、データファイルを参照したりする際には、それらが外部記憶部に記憶されている場合は、一旦、それらの情報を内部記憶部に取り込まなければならないものがある。

このような情報処理装置においては、ユーザーはアプリケーションの起動やデータファイルの参照などの際において、それらの情報が内部記憶部に記憶されているのか外部記憶部に記憶されているのかを判断し、それに応じた操作を行わなければならない。例えばメモリカードなどの外部記憶部に記憶されているアプリケーションソフトウェアを起動する場合は、ユーザーはまずメモリカードからのアプリケーションソフトウェアのロードを指示する操作を行って、ロードが実行された後に、起動操作を行う必要がある。

このような操作は、ユーザーにとって煩雑であり、また、常に目的とする情報が内部記憶部か外部記憶部のどちらに記憶されているかを認識していなければならないため、操作が簡易ではなく、情報処理装置の使用性はよいものであるとは言えない。

【 0 0 0 4 】

【課題を解決するための手段】

本発明はこのような問題に鑑みて、ユーザーがアプリケーションソフトウェアやデータファイルが記憶されている部位が内部記憶部か外部記憶部を意識するこ

となく、かつ簡易な操作性で、内部記憶部及び外部記憶部に記憶されている情報を利用できるようにすることを目的とする。

【 0 0 0 5 】

このため本発明の情報処理装置は、アプリケーションソフトウェア又はデータファイルが記憶可能な内部記憶手段と、アプリケーションソフトウェア又はデータファイルが記憶可能な外部記憶部との間で情報の入出力を可能とする接続手段と、上記接続手段により接続された上記外部記憶部に記憶されているアプリケーションソフトウェア又はデータファイルを認識する認識手段と、上記内部記憶手段に記憶されたアプリケーションソフトウェア又はデータファイル、及び上記認識手段で認識されたアプリケーションソフトウェア又はデータファイルを提示すると共に、提示されたアプリケーションソフトウェアの起動又はデータファイルの使用を選択することができるようにされたユーザーインターフェース手段と、上記ユーザーインターフェース手段による操作により、上記外部記憶部に記憶されているアプリケーションソフトウェアの起動が求められた際に、そのアプリケーションソフトウェアを上記内部記憶手段にロードした上で起動処理を行う起動処理制御手段と、を備えるようにする。

また上記ユーザーインターフェース手段による操作により、もしくは起動中のアプリケーションソフトウェアの処理により、上記外部記憶部に記憶されているデータファイルの参照が求められた際に、そのデータファイルを上記内部記憶手段にロードした上で、参照処理が行われるようにする参照処理制御手段を、さらに備えるようにする。

ここで上記外部記憶部は可搬性記録媒体であり、上記接続手段は、可搬性記録媒体に対する記録再生部として実現されるようにする。

又は、上記外部記憶部は有線又は無線の通信路を介して接続される外部サーバであり、上記接続手段は、上記通信路を介して通信を行う通信部として実現されるようにする。

【 0 0 0 6 】

本発明の情報処理方法は、アプリケーションソフトウェアの起動操作を検知する起動検知手順と、上記起動検知手順で起動操作を検知したアプリケーションソ

フトウェアが内部記憶手段に記憶されているか、或いは接続された外部記憶部に記憶されているかを判別する判別手順と、上記判別手順において、上記外部記憶部に記憶されていると判別された場合は、上記外部記憶部からアプリケーションソフトウェアを上記内部記憶手段にロードするロード手順と、上記ロード手順により上記内部記憶手段にロードされたアプリケーションソフトウェアを起動する起動手順と、が行われるようにする。

またユーザーインターフェースによる操作により、もしくは起動中のアプリケーションソフトウェアの処理により、上記外部記憶部に記憶されているデータファイルの参照が求められた際に、そのデータファイルを上記内部記憶手段にロードした上で、参照処理が行われるようにするデータファイル参照手順が行われるようにする。

ここで上記外部記憶部は可搬性記録媒体であり、上記ロード手順では、可搬性記録媒体に対する再生動作を行ってロードを実行する。

又は上記外部記憶部は有線又は無線の通信路を介して接続される外部サーバであり、上記ロード手順は、上記通信路を介して通信を行ってロードを実行する。

【0007】

即ち本発明は、或るアプリケーションソフトウェアの起動操作が行われた場合において、そのアプリケーションソフトウェアが外部記憶部に記憶されていた場合、そのアプリケーションソフトウェアが外部記憶部から内部記憶手段にロードされてから起動されるようにすることで、ユーザーがロード操作を行わなくてもよいようにする。また、これによってユーザーはアプリケーションソフトウェアが内部記憶手段に記憶されているか外部記憶部に記憶されているかを意識しなくてもよいものとなる。データファイルについても同様である。

【0008】

【発明の実施の形態】

以下、本発明の実施の形態を次の順序で説明する。この実施の形態の情報処理装置は、いわゆるPDA機器としての情報処理装置である。また外部記憶部の一例としてメモリカード及び通信網を介して接続されるサーバを挙げる。

1. 情報処理装置の外観例
2. 情報処理装置の構成
3. OS構造及びデータベース構造
4. メモリカード
 - 4-1 外観
 - 4-2 メモリカードの端子及び内部構造
 - 4-3 ファイルシステム処理階層
 - 4-4 物理的データ構造
 - 4-5 物理アドレス及び論理アドレスの概念
 - 4-6 論理-物理アドレス変換テーブル
 - 4-7 ディレクトリ構造
5. FAT構造
6. メモリカードと情報処理装置のインターフェース
7. アプリケーションソフトウェアの起動及びデータ参照処理
8. 通信網を介した外部サーバの例

【0009】

1. 情報処理装置の外観例

本例の情報処理装置の外観例を図1に示す。

この情報処理装置1は、いわゆるPDA機器として携帯に適した小型軽量の装置とされる。また記録媒体として、後述するメモリカード70を装着し、記録再生を行うことができるものとする。

なお本発明としては、携帯型の情報処理装置に限られず、パーソナルコンピュータをはじめとするあらゆるタイプの情報処理装置に適用できるものであり、また装置が記録を行う記録媒体はメモリカードに限られず、HDD、光ディスク、光磁気ディスク、或いは装置内に固定的に配置されるRAM、フラッシュメモリなど、他の種の記録媒体であってもよいものである。

【 0 0 1 0 】

図 1 (a) (b) (c) (d) は情報処理装置 1 の外観例としての平面図、右側面図、左側面図、上面図を示している。

図 1 (d) に示すように装置上面側には後述するメモリカード 7 0 を装着可能なメモリスロット 7 が形成されており、この情報処理装置 1 は、メモリスロット 7 に装着されたメモリカード 7 0 に対する各種データ（コンピュータ用データ、音楽データ、音声データ、動画像データ、静止画像データ、制御データなど）の記録再生が可能とされる。

なお、この図 1 の例ではメモリスロット 7 が 2 つ形成されていることから、2 つのメモリカード 7 0 を同時に装着できるようになされている。もちろん、形成するメモリスロット 7 の数は 1 つでもよいし、3 つ以上でもよい。

【 0 0 1 1 】

この情報処理装置 1 には、平面上に例えば液晶パネルによる表示部 2 が形成され、アプリケーションソフトウェアの起動及び各種処理に伴う画像、データとしての画像や文字、再生される音声、音楽に付随する情報、さらには操作のガイドメッセージ、再生や編集操作等のためのメニュー画面などが表示される。

【 0 0 1 2 】

情報処理装置 1 上には、ユーザーの操作のための各種の操作子が設けられる。例えば操作キー 3 a、ジョグダイヤル 3 b、プッシュダイヤル 3 c などがそれぞれ所要部位に形成される。

これらの操作子によりユーザーは、例えば電源操作、メニュー操作、選択操作、文字等の入力操作、その他必要とされる各種の操作を行うことができる。

これらの操作子はもちろん一例にすぎない。即ち配備する操作子の数、種類、位置は多様に考えられる。

【 0 0 1 3 】

また、情報処理装置 1 上には、スピーカ 4、マイクロホン 5、撮像部 6 も形成され、音声の出力、入力、撮像による画像の取込なども実行できるようにされている。

【 0 0 1 4 】

また各種機器との接続のために、各種端子が形成される。例えば図 1 (b) のように、ヘッドホン端子 10、ライン出力端子 12、ライン入力端子 11 などが形成され、また図 1 (c) のように I E E E 1 3 9 4 端子 8、U S B (universal serial bus) 端子 9 などが形成される。

なお、これらの端子の種類、数、配置位置も、他の例が多様に考えられる。

例えば光ケーブル対応のデジタル入出力端子を備えるようにしたり、或いは S C S I コネクタ、シリアルポート、R S 2 3 2 C コネクタなどが形成されるようにしても良い。

【 0 0 1 5 】

また、この情報処理装置の場合は、例えば公衆回線等の通信網を介して所定のサーバやインターネット等から情報のダウンロードが可能とされるものとし、アンテナ 13 として、通信網の基地局との間の無線通信を行うためのアンテナが設けられる。

【 0 0 1 6 】

2. 情報処理装置の構成

図 2 に情報処理装置 1 の内部構成を示す。

図示するように情報処理装置 1 内には、まず中核となる部位として、システムコントローラ 21、CPU 22、フラッシュROM 23、D-RAM 24 が設けられる。また基本的なユーザーインターフェースのための部位として操作部 35、表示制御部 27、表示部 2 が形成される。

【 0 0 1 7 】

システムコントローラ 21 は操作部 35 からの操作情報を入力し、それに応じて CPU 22 に割り込みをかける。

操作部 35 とは、図 1 に示した各種操作子 3a、3b、3c に相当する。また図 1 では説明しなかったが、表示部 2 に操作キーやアイコンの表示を行うとともに

に表示部 2 上でのタッチ検出機構を設けることで、タッチパネル操作子を形成してもよく、その場合のタッチパネル操作子も図 2 でいう操作部 3 5 に含まれるものとなる。

【0018】

CPU 2 2 は基本ソフト (OS : Operating System) やアプリケーションプログラムが動作される部位となる。

CPU 2 2 はシステムコントローラ 2 1 を介して供給される操作情報に応じて所要の処理を実行する。

フラッシュROM 2 3 は、基本動作プログラム、各種処理定数、設定情報などを記憶する領域とされる。

D-RAM 2 4 は、各種処理に必要な情報の記憶、データのバッファリング、CPU 2 2 のワークエリアの拡張、その他、CPU 2 2 の処理に応じて多様に使用される。また D-RAM 2 4 にはストレージエリア (不揮発性領域) が設けられており、そのストレージエリアには OS やアプリケーションソフトウェアがインストールされる。

そして D-RAM 2 4 にインストールされたアプリケーションソフトウェアは、ユーザからの操作に応じて起動され、CPU 2 2 により実行される。

またアプリケーションソフトウェアはユーザーインターフェース画面を持ち、ユーザーの指示による状態遷移に基づいて、D-RAM 2 4 に確保されたフレームバッファに描画を行う。

描画された画像データは、表示制御部 2 7 に送られ、表示部 2 に表示される。

【0019】

また上述したようにメモ리카ード 7 0 に対するメモリスロット 7 が形成され、メモ리카ード 7 0 を装着できるが、CPU 2 2 は、メモ리카ードインターフェース 2 8 を介して装着されたメモ리카ード 7 0 に対して書込又は読み出しアクセスすることができる。メモ리카ードインターフェース 2 8 とメモ리카ード 7 0 との間のインターフェース動作については後述する。

CPU 2 2 は、装着されたメモ리카ード 7 0 を、拡張的なメモリ領域として利用することができる。

また、もちろんメモ리카ード70にアプリケーションプログラムが記録されていれば、それをD-RAM24にインストールしたり、或いはアプリケーションやデータをD-RAM24にロードすることで、所要処理を実行させることができる。

また、或るアプリケーションに基づいてCPU22が、作成した文書データ、画像データ、オーディオデータ、表計算データなどを、メモ리카ード70に記録することもできる。

なお、メモリスロット7にメモ리카ード70が装着されたことを検出することで、メモ리카ード70に対する動作が記録再生動作可能になったり、或いはメモ리카ード70に記録されているアプリケーションやデータが自動的にD-RAM24に展開されるなどの、いわゆるホットプラグイン動作も可能である。

またメモ리카ードインターフェース28は、メモ리카ード70に記録するデータについての暗号化処理や、読み出したデータの暗号解読処理なども可能とされる。

【0020】

撮像部6は例えばCCD撮像素子及び撮像回路系によりに形成される。撮像部6により取り込まれた撮像画像データは、撮像データインターフェース34を介してD-RAM24に取り込むことができ、またCPU22は所定のアプリケーションプログラムに基づく動作により、撮像画像データの編集やメモ리카ード70への記録等を実行できる。

【0021】

オーディオインターフェース29は、上述したスピーカ4、マイクロホン5、ヘッドホン端子10、ライン出力端子12、ライン入力端子11から入出力されるオーディオデータのインターフェース部位となる。

例えばマイクロホン5或いはライン入力端子11から入力されたアナログオーディオ信号は、入力オーディオ処理部32でそれぞれ所定の増幅処理やフィルタリングが行われ、A/D変換器33でデジタルオーディオデータとされてオーディオインターフェース29に供給される。オーディオインターフェース29は、入力されたデジタルオーディオデータについて、CPU22の制御に基づいて処

理や出力を実行する。例えば所要の圧縮エンコード処理を行った後、メモリカードインターフェース 28 に供給し、メモリカード 70 に記録させることができる。またオーディオインターフェース 29 は、例えばメモリカード 70 から読み出されるなどして供給されたデジタルオーディオデータについて所定のデコード処理を行い、D/A変換器 30 に供給する。D/A変換器 30 はデジタルオーディオデータをアナログオーディオ信号に変換する。出力オーディオ処理部 31 は供給されたアナログオーディオ信号について、出力先に応じた所定の増幅処理、インピーダンス調整などを行い、スピーカ 4、ヘッドホン端子 10、ライン出力端子 12 に出力する。

【0022】

USBインターフェース 25 は、USBコネクタ 9 に接続された外部機器との間の通信インターフェースである。CPU 22 はUSBインターフェース 25 を介して外部のパーソナルコンピュータ或いは周辺機器などとの間でデータ通信を行うことができる。例えばこの情報処理装置 1 で扱われるアプリケーションソフトウェアや、データファイルとしての画像データ、オーディオデータなどの送受信が実行される。

同様にIEEE1394インターフェース 26 は、IEEE1394端子 8 に接続された外部機器との間の通信インターフェースである。CPU 22 はIEEE1394インターフェース 26 を介して外部の情報機器との間で各種データ通信を行うことができる。

【0023】

通信部 36 は、例えば後に図 24 で述べる通信網を介して、外部のサーバやインターネットホームページ等との間で情報通信を実行するための部位である。特に本例の場合は、通信部 36 を介して外部のサーバ等からアプリケーションソフトウェアやデータファイルをダウンロードできるようにされているものである。

なお、ここではこの情報処理装置 1 が通信網に対して無線通信端末であるとしての例で述べているが、例えばモデムや接続端子を設けて、有線で公衆電話回線などに接続できるようにしてもよい。

【 0 0 2 4 】

また本例の情報処理装置 1 がアプリケーションソフトウェアやデータファイルをダウンロード可能に接続されえるのは公衆回線を介したサーバ等のみでなく、専用回線で構築された通信システム内の端末或いはサーバシステム等であってもよい。

【 0 0 2 5 】

なお、この図 2 に示す情報処理装置 1 の構成はあくまでも一例であり、これに限定されるものではない。即ち、一般にパーソナルコンピュータや P D A 機器で採用されている各種構成部位を追加したり、或いは実際の製品として不要の部位を削除することは、設計上の都合により決められるものである。

【 0 0 2 6 】

3. O S 構造及びデータベース構造

続いて図 3 で、本例の情報処理装置 1 に搭載される O S 構造について説明する。図 3 に示すように、O S は、基本ソフトの中心部分としてのカーネルを含むマネージャ層と、標準ライブラリ、及び制御 I C などのハードウェアのレイヤとなる H A L (Hardware Abstraction Layer) から構成される。

アプリケーションソフトウェアは、このような O S 構造による基本動作上で動作される。

また H A L に対しては、1 又は複数のデバイスドライバとして階層が付加され実際のハードウェア (H W) が駆動される。

【 0 0 2 7 】

ここで、特に本例の情報処理装置 1 の場合は、メモ리카ード 7 0 をドライブ可能とし、かつ後述するがメモ리카ード 7 0 のデータは F A T により管理されることから、O S に F A T ライブラリが付加され、さらに、メモ리카ードをハンドリングするためのライブラリ (M S ライブラリ) が付加される。

そしてこの F A T ライブラリ及び M S ライブラリに基づいて、メモリドライブ

がメモ리카ード70がドライブされる構造とされている。

【0028】

このようなOS構造を持つ本例の情報処理装置1では、さらに通常でいうところの「ファイル」に相当する概念として、「データベース」という概念が導入されている。

ここでいう「データベース」とは、通常いうところのデータベースのように単にデータを蓄積していったものではなく、データベース自体がデータを管理できる構造としてフォーマット化されている。この意味で、「データベース」は「ファイル」に相当する。

【0029】

図4にデータベース構造を示す。即ちデータベースには、ヘッダ(DTBヘッダ)としてデータベースネーム(DTB Name)及びその他情報を含む領域が形成され、さらにポインタテーブルが配される。そしてデータ領域に記録される実際のデータは、ポインタテーブルに記録されたポイント情報により、位置的な管理が行われる状態となっている。

【0030】

このような構造のデータベースとしては、2種類のものが存在する。例えば一般に1つのアプリケーションソフトウェアは複数のファイルで構成され、その中には実行ファイル(***.exe)と、データファイル(***.data)があるが、その実行ファイル(***.exe)に相当するものとして「リソースデータベース(***.prc)」があり、またデータファイル(***.data)に相当するものとして「データベースデータベース(***.dtb)」がある。

【0031】

本例の情報処理装置1では、このような「データベース」という概念によりデータを扱う。従って、メモ리카ード70において記録再生されるファイル(FATで扱われるファイル)も、上記データベースの形態となる。

なお本明細書では、「ファイル」という言葉を用いるが、これは一般的な概念にあわせて用いているものであり、本実施の形態に関していえば、「ファイル」

とは上記構造のデータベースの意味となる。

【 0 0 3 2 】

4. メモリカード

4 - 1 外観

次にメモリカード 7 0 について説明していく。

まず図 5 にメモリカード 7 0 の外形形状を示す。

メモリカード 7 0 は、例えば図 5 に示すような板状の筐体内部に例えば所定容量のメモリ素子を備える。本例としては、このメモリ素子としてフラッシュメモリ (Flash Memory) が用いられるものである。

図 5 に平面図、正面図、側面図、底面図として示す筐体は例えばプラスチックモールドにより形成され、サイズの具体例としては、図に示す幅 $W 1 1$ 、 $W 1 2$ 、 $W 1 3$ のそれぞれが、 $W 1 1 = 6 0 \text{ mm}$ 、 $W 1 2 = 2 0 \text{ mm}$ 、 $W 1 3 = 2 . 8 \text{ mm}$ となる。

【 0 0 3 3 】

筐体の正面下部から底面側にかけて例えば 1 0 個の電極を持つ端子部 7 2 が形成されており、この端子部 7 2 から、内部のメモリ素子に対する読出又は書込動作が行われる。

筐体の平面方向の左上部は切欠部 7 3 とされる。この切欠部 7 3 は、このメモリカード 7 0 を、例えばドライブ装置本体側の着脱機構へ装填する際などに挿入方向を誤ることを防止するためのものとなる。

また筐体上面から底面側にかけて、ラベル貼付面 7 4 が形成され、ユーザーが記憶内容を書いたラベルを貼付できるようにされている。

さらに底面側には、記録内容の誤消去を防止する目的のスライドスイッチ 7 5 が形成されている。

【 0 0 3 4 】

このようなメモリカード 7 0 においては、フラッシュメモリ容量としては、4

MB (メガバイト), 8MB, 16MB, 32MB, 64MB, 128MBの何れかであるものとして規定されている。

またデータ記録/再生のためのファイルシステムとして、いわゆるFAT (File Allocation Table) システムが用いられている。

【0035】

書込速度は1500KByte/sec~330KByte/sec、読出速度は2.45MByte/secとされ、書込単位は512バイト、消去ブロックサイズは8KB又は16KBとされる。

また電源電圧Vccは2.7~3.6V、シリアルクロックSCLKは最高20MHzとされる。

【0036】

4-2 メモリカードの端子及び内部構造

図6に端子部72の電極構造を示す。図5に示したように端子部72は10個の平面電極が1列に並んだ構造とされるが、図6に示すように各電極(端子T1~T10)は次の通りとなる。

【0037】

端子T1及びT10は検出電圧Vss端子とされる。

端子T2は、シリアルプロトコルバスステート信号BSの入力端子とされる。

端子T3及びT9は電源電圧Vcc端子とされる。

端子T4はデータ端子、つまりシリアルプロトコルデータ信号の入出力端子とされる。

端子T5及びT7はリザーブ(予備)とされる。

端子T6は検出端子とされ、ドライブ装置側(情報処理装置1のメモリカードインターフェース)がメモリカードの装着検出に用いる。

端子T8は、シリアルクロックSCLKの入力端子とされる。

【 0 0 3 8 】

また図 6 にはメモリカード 7 0 の内部構成も示している。

メモリカード 7 0 の内部は、コントロール IC 8 0 とフラッシュメモリ 8 1 が設けられている。コントロール IC 8 0 はフラッシュメモリ 8 1 に対する書込／読出動作を実行する部位となる。

図からわかるように、コントロール IC 8 0 に対しては、端子 T 2 からのシリアルプロトコルバスステート信号 B S、端子 T 8 からのシリアルクロック S C L K が供給される。書込動作時には、コントロール IC 8 0 は、これらのシリアルプロトコルバスステート信号 B S、シリアルクロック S C L K に従って、端子 T 4 から供給されるデータのフラッシュメモリ 8 1 への書込を行う。また読出時には、シリアルプロトコルバスステート信号 B S、シリアルクロック S C L K に従って、フラッシュメモリ 8 1 からデータを読み出し、端子 T 4 からドライブ装置側に出力する。

【 0 0 3 9 】

また検出電圧 V_{ss} は、検出端子 T 6 に供給されており、ドライブ装置側では、図示するように抵抗 R によって検出端子 T 6 の端子電圧を検出することで、このメモリカード 7 0 が装着部（メモリスロット 7）に接続されているか否かを検出できるようにされる。

【 0 0 4 0 】

4 - 3 ファイルシステム処理階層

続いて、メモリカード 7 0 を記録媒体とするシステムにおけるフォーマットについて説明していく。

図 7 は、メモリカード 7 0 を記録媒体とするシステムのファイルシステム処理階層を示すものである。

この図に示すように、ファイルシステム処理階層としては、アプリケーション処理層の下に、順次、ファイル管理処理層、論理アドレス層、物理アドレス層、

フラッシュメモリアクセスがおかれる。

この階層では、ファイル管理処理層がいわゆる F A T (File Allocation Table) となる。

また、この図から分かるように、本例のファイルシステムでは論理アドレス及び物理アドレスという概念が導入されているが、これについては後述する。

【 0 0 4 1 】

4 - 4 物理的データ構造

図 8 には、メモ리카ード 7 0 内の記憶素子である、フラッシュメモリ 8 1 の物理的データ構造が示されている。

フラッシュメモリ 8 1 としての記憶領域は、セグメントという固定長のデータ単位が大元となる。このセグメントは、1 セグメントあたり 4 MB (メガバイト) 或いは 8 MB として規定されるサイズであり、1 つのフラッシュメモリ 8 1 内におけるセグメント数は、そのフラッシュメモリ 8 1 の容量に依存して異なってくる。

【 0 0 4 2 】

そして、この 1 セグメントを図 8 (a) に示すように、ブロックという固定長のデータ単位として 8 K B (キロバイト) 又は 1 6 K B により区切るようにされる。原則として、1 セグメントは 5 1 2 ブロックに区切られることから、図 8 (a) に示すブロック n については、 $n = 5 1 1$ とされることになる。但し、フラッシュメモリ 8 1 では、書き込み不可な損傷エリアであるディフェクトエリアとしてのブロック数が所定数の範囲で許可されているため、データ書き込みが有効とされる実質的なブロック数を対象とすれば、上記 n は 5 1 1 よりも少なくなる。

【 0 0 4 3 】

図 8 (a) に示すようにして形成されるブロック 0 ~ n のうち、先頭の 2 つのブロック 0, 1 はブートブロックといわれる。但し、実際には有効なブロックの

先頭から2つのブロックがブートブロックとして規定されることになっており、必ずしもブートブロックがブロック0, 1である保証はない。

そして、残りのブロックが、ユーザデータが格納されるユーザブロックとなる。

【0044】

1ブロックは、図8(d)に示すようにして、ページ0～mにより分割される。1ページの容量は、図8(e)に示すように、512バイトのデータエリアと16バイトの冗長部よりなる、528(=512+16)バイトの固定長とされる。なお、冗長部の構造については図8(f)により後述する。

また、1ブロック内のページ数としては、1ブロックの容量が8KBの場合には16ページ、16KBの場合には32ページとなる。

【0045】

このような、図8(d)(e)に示されるブロック内のページ構造は、上記ブートブロックとユーザブロックとで共通である。

また、フラッシュメモリ81では、データの読み出し、及び書き込みはページ単位で行われ、データの消去はブロック単位で行われるものとされる。そして、データの書き込みは、消去済みのページに対してしか行われないものとされている。従って、実際のデータの書き換えや書き込みは、ブロック単位を対象として行われることになる。

【0046】

先頭のブートブロックは、図8(b)に示すように、ページ0に対してヘッダーが格納され、ページ1には初期不良データの位置(アドレス)を示す情報が格納される。また、ページ2にはCIS/IDSといわれる情報が格納される。

2つめのブートブロックは図8(c)に示すように、ブートブロックとしてのバックアップのための領域とされている。

【0047】

図8(e)に示された冗長部(16バイト)は、図8(f)に示す構造を有する。

この冗長部は、図のように先頭の第0バイト～第2バイトの3バイトが、デー

タエリアのデータ内容の更新に応じて書き換えが可能なオーバーライトエリアとされる。このオーバーライトエリアのうち、第0バイトにはブロックステータスが格納され、第1バイトにはデータステータスが格納される(Block Flag Data)。また、第2バイトの上位の所定ビットを利用して変換テーブルフラグ(Page Data Status1)が格納される。

【0048】

原則として第3バイト～第15バイトは、その内容が現ページのデータ内容に応じて固定とされ、書き換えが不可とされる情報が格納される領域となる。

第3バイトにはアクセス許可やコピー禁止指定等を示す管理フラグ(Block Info)が格納される。

第4、第5バイトから成る2バイトの領域には、後述する論理アドレス(Logic Address)が格納される。

第6～第10バイトの5バイトの領域は、フォーマットリザーブの領域とされ、続く第11、第12バイトの2バイトが、上記フォーマットリザーブに対して誤り訂正を施すための分散情報ECCを格納する領域とされる。

残る第13～第15バイトには、図8(e)に示すデータエリアのデータに対して誤り訂正を行うためのデータECCが格納される。

【0049】

上記図8(f)に示した冗長部の第3バイトに格納される管理フラグは、図9に示すようにして、ビット7～ビット0の各ビットに、その内容が定義されている。

ビット7、6、及びビット1、0はリザーブ(未定義)領域とされている。

ビット5は現ブロックに対してのアクセス許可の「有効」(‘1’; Free) / 「無効」(‘0’; Read Protected)を示すフラグが格納される。ビット4には現ブロックについてのコピー禁止指定(‘1’; OK / ‘0’; NG)についてのフラグが格納される。

【0050】

ビット3は変換テーブルフラグとされる。この変換テーブルフラグは、現ブロックが後述する論理-物理アドレス変換テーブルであるのか否かを示す識別子で

あり、このビット3の値が‘0’とされていれば、現ブロックは論理－物理アドレス変換テーブルであることが識別され、‘0’であれば無効となる。つまり、現ブロックは論理－物理アドレス変換テーブルではないことが識別される。

【0051】

ビット2はシステムフラグが格納され、‘1’であれば現ブロックがユーザブロックであることが示され、‘0’であればブートブロックであることが示される。

【0052】

ここで、セグメント及びブロックと、フラッシュメモリ容量との関係を図13（左3列を参照）により説明しておく。

メモリカード70のフラッシュメモリ容量としては、4MB、8MB、16MB、32MB、64MB、128MBの何れかであるものとして規定されている。

そして、最も容量の小さい4MBの場合であると、1ブロックは8KBと規定された上で、そのブロック数としては512個とされる。つまり、4MBはちょうど1セグメントの容量を有するものとされる。そして、4MBの容量であれば、同様に1ブロック＝8KBの容量が規定された上で、2セグメント＝1024ブロックとなる。なお、前述したように、1ブロック＝8KBであれば、1ブロック内のページ数は16ページとなる。

但し16MBの容量では、1ブロックあたりの容量として8KBと16KBの両者が存在することが許可されている。このため、2048ブロック＝4セグメント（1ブロック＝8KB）のものと、1024ブロック＝2セグメント（1ブロック＝16KB）のものの2種類が在ることになる。1ブロック＝16KBの場合には、1ブロック内のページ数は32ページとなる。

【0053】

また、32MB、64MB、128MBの容量では、1ブロックあたりの容量は16KBのみであるとして規定される。従って、32MBでは2048ブロック＝4セグメントとなり、64MBでは4096ブロック＝8セグメントとなり、128MBでは8192ブロック＝16セグメントとなる。

【0054】

4-5 物理アドレス及び論理アドレスの概念

次に、上述したようなフラッシュメモリの物理的データ構造を踏まえたうえで、図10に示すデータ書き換え動作に従って、本例のファイルシステムにおける物理アドレスと論理アドレスの概念について説明する。

【0055】

図10(a)には、或るセグメント内から4つのブロックを抜き出して、これを模式的に示している。

各ブロックに対しては物理アドレスが付される。この物理アドレスはメモリにおけるブロックの物理的な配列順に従って決まるもので、或るブロックとこれに対応付けされた物理アドレスとの関係は不変となる。ここでは、図10(a)に示す4ブロックに対して、上から順に物理アドレスの値として、105, 106, 107, 108が付されている。なお、実際の物理アドレスは2バイトにより表現される。

【0056】

ここで、図10(a)に示すように、物理アドレス105, 106で示されるブロックがデータの記憶されている使用ブロックで、物理アドレス107, 108で示されるブロックがデータが消去（即ち、未記録領域）された未使用ブロックとなっている状態であるとする。

【0057】

そして、論理アドレスであるが、この論理アドレスは、ブロックに対して書き込まれたデータに付随するようにして割り振られるアドレスとされる。そして、この論理アドレスが、後述するFATファイルシステムが利用するアドレスとされている。

図10(a)では、4つの各ブロックに対して、上から順に論理アドレスの値として、102, 103, 104, 105が付されている状態が示されている。

なお、論理アドレスも実際には2バイトにより表現されるものである。

【0058】

ここで、上記図10(a)に示す状態から、例えば物理アドレス105に格納されているデータの更新として、内容の書き換え又は一部消去を行うとする。

このような場合、フラッシュメモリのファイルシステムでは、同じブロックに対して更新したデータを再度書き込むことはせずに、未使用のブロックに対してその更新したデータを書き込むようにされる。

つまり、例えば図10(b)に示すようにして、物理アドレス105のデータは消去したうえで、更新されたデータはこれまで未使用ブロックであった物理アドレス107で示されるブロックに書き込むようにされる(処理①)。

【0059】

そして、処理②として示すように、データ更新前(図10(a))の状態では物理アドレス105に対応していた論理アドレス102が、更新されたデータが書き込まれたブロックの物理アドレス107に対応するように、論理アドレスについての変更を行うものである。これに伴って、データ更新前は物理アドレス107に対応していた論理アドレス104については、物理アドレス105に対応するように変更されている。

【0060】

つまり、物理アドレスはブロックに対して固有に付されるアドレスであり、論理アドレスは、一旦ブロックに対して書き込まれたデータに付随するようにしてついて回る、ブロック単位の書き込みデータに固有となるアドレスであるとみることができる。

【0061】

このようなブロックのスワップ処理が行われることで、或る同一の記憶領域(ブロック)に対して繰り返し集中的にアクセスされることが無くなり、書き換え回数の上限があるフラッシュメモリの寿命を延ばすことが可能となる。

そして、この際に論理アドレスを上記処理②のようにして扱うことで、ブロックのスワップ処理によって更新前と更新後のデータとで書き込まれるブロックの移動があるようにされても、FATからは同一のアドレスが見えることになり、

以降のアクセスを適正に実行することができるものである。

なお、後述する論理－物理アドレス変換テーブル上での更新のための管理を簡略にすることなどを目的として、ブロックのスワップ処理は、1セグメント内で完結するものとして規定されている。逆に言えば、ブロックのスワップ処理はセグメント間で跨るようには行われぬ。

【 0 0 6 2 】

4 - 6 論理－物理アドレス変換テーブル

上記図 1 0 による説明から分かるように、ブロックのスワップ処理が行われることで、物理アドレスと論理アドレスの対応は変化する。従って、フラッシュメモリに対するデータの書き込み及び読み出しのためのアクセスを実現するには、物理アドレスと論理アドレスとの対応が示される論理－物理アドレス変換テーブルが必要となる。つまり、論理－物理アドレス変換テーブルを F A T が参照することで、F A T が指定した論理アドレスに対応する物理アドレスが特定され、この特定された物理アドレスにより示されるブロックにアクセスすることが可能になるものである。逆に言えば、論理－物理アドレス変換テーブルが無ければ、F A T によるフラッシュメモリへのアクセスが不可能となる。

【 0 0 6 3 】

従来では、例えばセット本体に対してメモリカード 7 0 が装着されたときに、セット本体側のマイクロプロセッサがメモリカード 7 0 の記憶内容をチェックすることで、セット本体側で論理－物理アドレス変換テーブルの構築を行い、更にこの構築された論理－物理アドレス変換テーブルをセット本体側の R A M に格納するようになっていた。つまり、メモリカード 7 0 内には、論理－物理アドレス変換テーブルの情報は格納されてはいなかった。

これに対して本例では、以降説明するようにメモリカード 7 0 に対して、論理－物理アドレス変換テーブルを格納するように構成している。

【 0 0 6 4 】

図 1 1 は、本例のメモリカード 7 0 に対して格納される論理－物理アドレス変換テーブルの構築形態を概念的に示すものである。

つまり、本例では、例えば論理アドレスの昇順に従って、これに対応する 2 バイトの物理アドレスを格納するようにしたテーブル情報を論理－物理アドレス変換テーブルとして構築するようにされる。

なお、前述したように、物理アドレス、及び論理アドレスは共に 2 バイトで表現される。これは、1 2 8 MB の最大容量のフラッシュメモリの場合には 8 1 9 2 個のブロックが存在するため、最大で、この 8 1 9 2 個のブロック数をカバーできるだけのビット数が必要とされることに基づく。

このため、図 1 1 において例示している物理アドレスと論理アドレスとについても、実際に即して 2 バイトで表現している。

但し、ここでは、この 2 バイトを 1 6 進数により表記している。つまり、「0 x」によりその後続く値が 1 6 進法表記であることが示される。なお、この「0 x」により 1 6 進数であることを表す表記は、以降の説明において 1 6 進数を表記する場合にも同様に用いることとする。（但し、表記の煩雑化を防ぐため「0 x」を省略している図面もある。）

【 0 0 6 5 】

図 1 2 に、上記図 1 1 に示した概念に基づく論理－物理アドレス変換テーブルの構造例を示す。

論理－物理アドレス変換テーブルは、フラッシュメモリの最後のセグメント内の或るブロックに対して、図 1 2 に示すようにして格納される。

先ず図 1 2 (a) に示すように、ブロックを分割するページのうち、ページ 0 , 1 からなる 2 ページの領域がセグメント 0 用の論理－物理アドレス変換テーブルとして割り当てられる。例えば、図 1 3 にて説明したように、フラッシュメモリが 4 MB の容量であれば 1 セグメントしか有さないために、このページ 0 , 1 のみの領域が論理－物理アドレス変換テーブルの領域となる。

また、例えばフラッシュメモリが 8 MB の容量であれば 2 セグメントを有するため、セグメント 0 用の論理－物理アドレス変換テーブルとして割り当てられる

ページ 0, 1 に加え、これに続くページ 2, 3 の 2 ページがセグメント 1 用の論理－物理アドレス変換テーブルとして割り当てられることになる。

【 0 0 6 6 】

以降、フラッシュメモリの容量の増加に応じて、続く 2 ページごとにセグメントごとの論理－物理アドレス変換テーブルの割り当て領域が設定されていくことになる。そして、最大の 1 2 8 MB の容量を有する場合であれば 1 6 セグメントが存在するため、最大では、セグメント 1 5 用までのページが論理－物理アドレス変換テーブルの領域として割り当てられることになる。従って、最大の 1 2 8 MB の容量のフラッシュメモリでは、3 0 ページが使用されることになり、図 1 2 (a) に示すページ N としては、最大で $N = 2 9$ となる。

これまでの説明から分かるように、論理－物理アドレス変換テーブルは、セグメントごとに管理されるものである。

【 0 0 6 7 】

図 1 2 (b) は、1 セグメントあたりの論理－物理アドレス変換テーブルの構造を示すものとして、2 ページ分のデータエリアを抜き出して示している。つまり、1 ページのデータエリアは 5 1 2 バイト (図 8 (e) 参照) であることから、図 1 2 (b) には、1 0 2 4 ($= 5 1 2 \times 2$) バイトが展開されている状態が示されている。

【 0 0 6 8 】

図 1 2 (b) に示すように、この 2 ページ分のデータエリアである 1 0 2 4 バイトについて 2 バイトごとに区切り、この 2 バイトごとの領域を、先頭から順次、論理アドレス 0 用、論理アドレス 1 用 、のようにして割付を行い、最後は先頭から 9 9 1 バイト目と 9 9 2 バイト目の 2 バイトの領域を論理アドレス 4 9 5 用の領域として割り付けるように規定を行う。これら 2 バイトごとの領域に対して、各論理アドレスが対応する物理アドレスを書き込むようにする。従って、本例の論理－物理アドレス変換テーブルでは、実際のデータ更新によるブロックのスワップ処理などにより物理アドレスと論理アドレスの対応が変更された場合には、論理アドレスを基準として、物理アドレスの格納状態が更新されるようにしてテーブル情報の書き換えが行われることになる。

【 0 0 6 9 】

また、残る 9 9 3 バイト目から最後の 1 0 2 4 バイト目までの計 3 2 バイトの領域は、余剰ブロックの物理アドレスが格納される領域として割り当てられる。つまり、1 6 個の余剰ブロックの物理アドレスを管理することができる。ここでいう余剰ブロックとは、例えばブロック単位でデータの更新を行う際に書き換え対象となるデータを一時待避させる領域として設定されたいわゆるワークブロックなどを言うものである。

【 0 0 7 0 】

ところで、1 セグメントは 5 1 2 ブロックに分割されているものであると先に説明したのにも関わらず、図 1 2 に示したテーブル構造では、管理可能なブロック数が論理アドレス 0 用～論理アドレス 4 9 5 用の 4 9 6 ブロックとしている。これは、實際上、上記した余剰アドレスが設定されることと、前述したように、フラッシュメモリでは、或ブロック数のディフェクト（使用不可領域）が許可されている。そのため現実には、相当数のディフェクトブロックが存在することに依る。

従って、実際には、書き込み／消去が有効なブロックを管理するのに、4 9 6 ブロックを管理できるように構成しておけば充分とされるものである。

【 0 0 7 1 】

そして、このようにして論理－物理アドレス変換テーブルが格納されるブロックについては、これを形成する各ページの冗長部における管理フラグ（図 9 参照）のデータ内容として、この管理フラグのビット 3 に対して‘0’がセットされることになる。これにより、当該ブロックが論理－物理アドレス変換テーブルが格納されているブロックであることが示されることになる。

【 0 0 7 2 】

論理－物理アドレス変換テーブルが格納されるブロックも、論理－物理アドレス変換テーブルの内容の書き換えがあった場合には、例外なく、先に図 1 0 にて説明したスワップ処理が行われる。従って、論理－物理アドレス変換テーブルが記録されているブロックは不定であり、或る特定のブロックに論理－物理アドレス変換テーブルを格納するように規定することは出来ない。

そこで、F A Tは、フラッシュメモリにアクセスして上記した管理フラグのビット3が‘0’とされているブロックを検索することで、論理－物理アドレス変換テーブルが格納されているブロックを識別するようにされる。但し、論理－物理アドレス変換テーブルが格納されているブロックの検索がF A Tによって容易に行われるようにすることを考慮して、論理－物理アドレス変換テーブルが格納されているブロックは、そのフラッシュメモリ内における最後のナンバが付されたセグメントに在るように、本例では規定するものとされる。これにより、F A Tは最後のナンバが付されたセグメントのブロックのサーチだけで、論理－物理アドレス変換テーブルを検索することができる。つまり、論理－物理アドレス変換テーブルを検索するのに、フラッシュメモリの全てのセグメントを検索する必要は無いようにされる。

上記図12に示した論理－物理アドレス変換テーブルは、例えばメモリカード70の製造時において格納するようにされる。

【0073】

ここで、再度図13を参照して、フラッシュメモリ容量と論理－物理アドレス変換テーブルのサイズとの関係を説明しておく。

上記図11にて説明したように、1セグメントを管理するための論理－物理アドレス変換テーブルのサイズは2ページ分の1024バイト、つまり1KBとなる。従って、図13の最右列に記されているように、フラッシュメモリが4MB（1セグメント）の容量では論理－物理アドレス変換テーブルは1KBのサイズとなる。また、フラッシュメモリの容量が8MB（2セグメント）では論理－物理アドレス変換テーブルは2KB（4ページ）となる。

また、フラッシュメモリの容量が16MBの場合、2048ブロック＝4セグメントのものでは論理－物理アドレス変換テーブルは4KB（8ページ）、1024ブロック＝2セグメントのものでは論理－物理アドレス変換テーブルは2KB（4ページ）となる。

そして、フラッシュメモリの容量が32MB（4セグメント）では論理－物理アドレス変換テーブルは4KB（8ページ）、フラッシュメモリの容量が64MB（8セグメント）では論理－物理アドレス変換テーブルは8KB（16ページ

）となり、フラッシュメモリの容量が最大の128MB（16セグメント）では論理－物理アドレス変換テーブルは16KB（32ページ）となる。

【0074】

4-7 ディレクトリ構造

メモ리카ード70に記録されるディレクトリ構成例を図14に示す。

メモ리카ード70で扱うことのできる主データとしては、コンピュータ用データ、動画データ、静止画データ、メッセージデータ、オーディオデータ、制御用データなどがあるが、このためディレクトリ構造としては、ルートディレクトリから、「VOICE」（メッセージ用ディレクトリ）、「DCIM」（静止画用ディレクトリ）、「MOxxxxnn」（動画用ディレクトリ）、「CONTROL」（制御用ディレクトリ）、「HIFI」（オーディオ用ディレクトリ）、「PM」（情報処理装置用ディレクトリ）が配される。

【0075】

そして図示していないが、各ディレクトリの下には、サブディレクトリやファイル（上述したデータベース）、フォルダ等が配され、いわゆるツリー構造の形態をとることになる。

なお、もちろんこのようなディレクトリ構成は一例にすぎず、実際には情報処理装置1等による記録状況や記録されるファイル種別などに応じてディレクトリ構造が形成される。

【0076】

5. FAT構造

図7のファイルシステム階層で説明したように、ファイル管理処理はFATにより行われることになる。

即ち図2に示した構成の情報処理装置1により、メモ리카ード70に対する記録再生（データ書込／読出）を実現するには、アプリケーション処理での要求に

伴ってFATによるファイル記憶位置管理が参照され、さらに上述した論理－物理アドレス変換が行われて実際のアクセスが行われることになる。

ここで、FATの構造について説明しておく。

【 0 0 7 7 】

図15はFATによる管理構造の概要を示している。

なお、本例ではFAT及び論理－物理アドレス変換テーブルはメモ리카ード70内に格納されることになるが、図15に示すFAT構造が、メモ리카ード70内での管理構造となるものである。

【 0 0 7 8 】

図示するようにFAT管理構造は、パーティションテーブル、空き領域、ブートセクタ、FAT、FATのコピー、ルートディレクトリ、データ領域から成る。

データ領域には、クラスタ2、クラスタ3・・・として単位データを示しているが、このクラスタとは、管理単位となるFATで扱う1データ単位である。

一般にFATでは、クラスタサイズは標準で4Kバイトとされるが、このクラスタサイズは512バイト～32Kバイトの間で2のべき乗の大きさをとることができる。

本例のメモ리카ード70では、上述したように1つのブロックが8Kバイト又は16Kバイトとされるが、1ブロック＝8Kバイトとされるメモ리카ード70の場合は、FATで扱うクラスタは8Kバイトとされる。また1ブロック＝16Kバイトとされるメモ리카ード70の場合は、FATで扱うクラスタは16Kバイトとされる。即ち、8Kバイト又は16KバイトがFAT管理上でのデータ単位であり、かつメモ리카ード70でのブロックとしての1つのデータ単位とされる。なお、従ってメモ리카ードからみれば、FATで扱われるクラスタサイズ＝そのメモ리카ードのブロックサイズとなる。このため、本例の以降の説明については、簡略化のためにブロック＝クラスタとして考えることとする。

【 0 0 7 9 】

そして図15左側にブロックナンバとして $x \cdots (x+m-1)$ 、 $(x+m)$ 、 $(x+m+1)$ 、 $(x+m+2) \cdots$ と示したが、例えばこのように各ブロッ

クにおいてF A T構造を構築する各種データは記憶されることになる。

なお、実際には必ずしもこのように物理的に連続する各ブロックに各情報が記憶されるものではない。

【 0 0 8 0 】

F A T構造において、まずパーティションテーブルには、F A Tパーティション（最大2 Gバイト）の先頭と終端のアドレスが記述されている。

ブート領域には、いわゆる1 2 b i t F A T、1 6 b i t F A Tの別や、F A T構造（大きさ、クラスタサイズ、各領域のサイズなど）が記述される。

【 0 0 8 1 】

F A Tは、後述するように各ファイルを構成するクラスタのリンク構造を示すテーブルとなり、またF A Tについては続く領域にコピーが記述される。

ルートディレクトリには、ファイル名、先頭クラスタ番号、各種属性が記述される。これらの記述は1つのファイルにつき3 2バイト使用される。

【 0 0 8 2 】

F A Tにおいては、F A Tのエントリとクラスタは1対1で対応しており、各クラスタのエントリにはリンク先、つまり後に続くクラスタの番号が記述される。つまり、複数のクラスタ（＝ブロック）で形成されている或るファイルについてみると、まずディレクトリによって先頭のクラスタ番号が示され、F A Tにおけるその先頭クラスタのエントリには、次のクラスタ番号が示される。さらに次のクラスタ番号のエントリには、さらに次のクラスタ番号が示される。このようにクラスタのリンクがF A Tに記述される。

【 0 0 8 3 】

図1 6はこのようなリンクの概念を模式的に示している（数値は1 6進値）。

例えば2つのファイル「M A I N . C」「F U N C . C」が存在するとすると、ディレクトリにはこの2つのファイルの先頭クラスタ番号が例えば「0 0 2」「0 0 4」と記述される。

そしてファイル「M A I N . C」については、クラスタ番号「0 0 2」のエントリに次のクラスタ番号「0 0 3」が記述され、またクラスタ番号「0 0 3」のエントリに次のクラスタ番号「0 0 6」が記述される。さらに、クラスタ番号0

06がこのファイル「MAIN. C」の最後のクラスタであるとする、クラスタ番号「006」のエントリには、最後のクラスタであることを示す「FFF」が記述される。

これによりファイル「MAIN. C」がクラスタ「002」→「003」→「006」という順番で記憶されている。即ち、仮にクラスタ番号とメモ리카ード70でのブロック番号が一致していると仮定すると、ファイル「MAIN. C」は、メモ리카ード70内でブロック「002」「003」「006」に記憶されていることが表現されている。（但し、FATで扱うクラスタは、上述のように論理アドレスで扱うものとなるため、ブロックの物理アドレスとそのまま一致するものではない）

【0084】

また同様にファイル「FUNC. C」については、FATにより、クラスタ「004」→「005」に記憶されていることが表現される。

【0085】

なお、未使用のブロックに対応するクラスタについては、そのエントリは「000」とされる。

【0086】

ところでルートディレクトリの領域に記憶される各ファイルのディレクトリにおいては、図16に示した先頭クラスタ番号だけでなく、例えば図17のように各種データが記述される。

即ちファイル名、拡張子、属性、変更時刻情報、変更日付情報、先頭クラスタ番号、ファイルサイズが、それぞれ図示するバイト数で記述される。

【0087】

また或るディレクトリの下層となるサブディレクトリについては、図15のルートディレクトリの領域ではなく、データ領域に記憶される。つまりサブディレクトリは、ディレクトリ構造を持つファイルとして扱われる。そしてサブディレクトリの場合はサイズは無制限とされ、また自分自身へのエントリと親ディレクトリへのエントリが必要になる。

【0088】

図18に、或るルートディレクトリ内にファイル「DIR1」（属性＝ディレクトリ：つまりサブディレクトリ）があり、さらにその中にファイル「DIR2」（属性＝ディレクトリ：つまりサブディレクトリ）があり、さらにその中にファイル「FILE」が存在する場合の構造例を示している。

つまりルートディレクトリの領域には、サブディレクトリであるファイル「DIR1」としての先頭クラスタ番号が示され、上述したFATにより、クラスタX、Y、Zがリンクされている状態となる。

この図からわかるように、サブディレクトリ「DIR1」「DIR2」についてはファイルとして扱われてFATのリンクに組み込まれる。

【0089】

6. メモリカードと情報処理装置のインターフェース

図19により、メモリカード70と情報処理装置1のメモリカードインターフェース28の間のシリアルインターフェースシステム構成を説明する。

メモリカード70内のコントロールIC80は、図19に示すようにフラッシュメモリコントローラ80a、レジスタ80b、ページバッファ80c、シリアルインターフェース80dとしての各ブロックを有するものとなっている。

【0090】

フラッシュメモリコントローラ80aは、レジスタ80bに設定されたパラメータに基づいて、フラッシュメモリ81とページバッファ80cの間でのデータ転送を行う。

そしてページバッファ80cにバッファリングされたデータはシリアルインターフェース80dを介して情報処理装置1のメモリカードインターフェース28側に転送され、また情報処理装置1のメモリカードインターフェース28から転送されてきたデータはシリアルインターフェース80dを介してページバッファ10cにバッファリングされる。

【 0 0 9 1 】

メモ리카ードインターフェース 2 8 側では、メモ리카ード 7 0 に対するインターフェース構造として、ファイルマネージャ 6 0、転送プロトコルインターフェース 6 1、シリアルインターフェース 6 2 を有する。

ファイルマネージャ 6 0 はメモ리카ード 7 0 のファイル管理を行う。例えば本例のシステムではメモ리카ード 7 0 内にはメインデータファイルの管理のための管理ファイルが記憶されているが、情報処理装置 1 は装填されたメモ리카ード 7 0 から管理ファイルをよみこんで CPU 2 2 がファイルマネージャ 6 0 を形成することになる。メモ리카ード 7 0 へのアクセスはファイルマネージャ 6 0 に従って実行される。

転送プロトコルインターフェース 6 1 は、レジスタ 8 0 b、ページバッファ 8 0 c へのアクセスを実行する。

シリアルインターフェース 6 2 は、メモ리카ード 7 0 との間の 3 つの信号線、即ち S C L K (シリアルクロック)、B S (バスステイト)、S D I O (シリアルデータ入出力) において、任意のデータ転送を行うためのプロトコルを規定する。

【 0 0 9 2 】

以上の構成における各部の動作により、情報処理装置 1 によるメモ리카ード 7 0 (フラッシュメモリ 8 1) に対する読出アクセス/書込アクセスが実行される。

【 0 0 9 3 】

7. アプリケーションソフトウェアの起動及びデータ参照処理

続いて本例の情報処理装置 1 による特徴的な動作となるアプリケーションソフトウェアの起動及びデータ参照処理について説明していく。

上述したように情報処理装置 1 には D - R A M 2 4、フラッシュメモリ 2 3 が設けられており、これらは情報処理装置 1 の内部記憶部として、アプリケーション

ンソフトウェアやデータファイルを格納することができる。

また情報処理装置 1 がアクセスできる外部記憶部としてはメモリカード 7 0 が用意されている。

なお、通信部 3 6 が設けられていることで、通信網を介したサーバやインターネットホームページ等にもアクセス可能であり、従ってそれらのサーバ等も、ここでいう外部記憶部に含めることができる。これらの外部のサーバ等に関しては図 2 4 を用いて後述する。

【 0 0 9 4 】

ユーザーは情報処理装置 1 において任意のアプリケーションソフトウェアを起動させることで、そのアプリケーションソフトウェアに基づく処理を実行させることができる。もちろんアプリケーションソフトウェアの動作上において各種のデータファイルを参照し、内容を確認したり編集することなども可能である。

ここで、アプリケーションソフトウェアやデータファイルの記憶場所を考えると、D-RAM 2 4 等の内部記憶部である場合と、メモリカード 7 0 等の外部記憶部である場合が考えられる。

【 0 0 9 5 】

この情報処理装置 1 は、外部記憶部に保持されたアプリケーションソフトウェアやデータファイルは、内部記憶部にロードした後でなければ実行或いは参照できないものとされているが、従ってこの種の情報処理装置では、従来は、上述したように外部記憶部のアプリケーションソフトウェアやデータファイルを利用する場合は、ユーザーは予めそれらの情報を内部記憶部にロードさせておかなければならなかった。

これに対して本例では、ユーザーはアプリケーションソフトウェアやデータファイルが内部記憶部に格納されているか外部記憶部に格納されているかを意識せずに、単に所要のアプリケーションソフトウェアを起動させたり参照するデータファイルを選択する操作を行うのみでよいようにするものである。

このため、概略的にいえば本例では図 2 0 に示すような動作が行われる。

【 0 0 9 6 】

図 2 0 は起動されるアプリケーションソフトウェアと、そのアプリケーション

ソフトウェアによる動作中に参照されるデータファイルについての格納場所のパターンを示したものである。

①は、起動されるアプリケーションソフトウェアが内部記憶部にインストールされていたものであって、また参照されるデータファイルも内部記憶部に格納されていた場合である。

この場合、CPU 22は、D-RAM 24に格納されているアプリケーションソフトウェアを通常に起動させる処理を行い、またデータファイルを参照する際には、D-RAM 24に格納されているデータファイルを読み出してアプリケーションソフトウェアによる処理に供することになる。

【0097】

②は、起動されるアプリケーションソフトウェアが内部記憶部にインストールされているが、参照されるデータファイルが外部記憶部に格納されていた場合である。

この場合アプリケーションソフトウェアの起動については、CPU 22は、D-RAM 24に格納されているアプリケーションソフトウェアを通常に起動させる処理を行なう。一方、そのアプリケーションソフトウェアによる動作中においてデータファイルを参照する際には、メモ리카ード70に格納されているデータファイルを読み出してD-RAM 24にロードさせた後において、アプリケーションソフトウェアによる処理に供することになる。

【0098】

③は、起動されるアプリケーションソフトウェアが外部記憶部に格納され、参照されるデータファイルが内部記憶部に格納されていた場合である。

この場合アプリケーションソフトウェアの起動については、CPU 22は、まずメモ리카ード70からアプリケーションソフトウェアをD-RAM 24にロードした後において、そのD-RAM 24にロードされたアプリケーションソフトウェアを起動させる処理を行なう。また、そのアプリケーションソフトウェアによる動作中においてデータファイルを参照する際には、D-RAM 24に格納されているデータファイルを読み出してアプリケーションソフトウェアによる処理に供することになる。

【 0 0 9 9 】

④は、起動されるアプリケーションソフトウェアが外部記憶部に格納され、参照されるデータファイルも外部記憶部に格納されていた場合である。

この場合アプリケーションソフトウェアの起動については、CPU 22は、まずメモリカード70からアプリケーションソフトウェアをD-RAM 24にロードした後において、そのD-RAM 24にロードされたアプリケーションソフトウェアを起動させる処理を行なう。また、そのアプリケーションソフトウェアによる動作中においてデータファイルを参照する際には、メモリカード70に格納されているデータファイルを読み出してD-RAM 24にロードさせた後において、アプリケーションソフトウェアによる処理に供することになる。

【 0 1 0 0 】

即ち本例の場合、アプリケーションソフトウェアやデータファイルが外部記憶部に格納されているものであった場合は、それらが必要とされた際において内部記憶部に自動的にロードして、起動や参照を実行する。つまりユーザがメモリカード70等に記憶されたアプリケーションソフトウェアやデータファイルを予めD-RAM 24に取り込ませるような操作を行うことを不要とするものである。

【 0 1 0 1 】

以上のような処理を実行するためのCPU 22の処理を図21、図22、図23で説明していく。

図21は、情報処理装置1が起動されてからの処理を示している。

ユーザーによって起動（電源投入）されることに応じて、CPU 22はステップF101からF102に進み、所定の起動処理を実行する。

【 0 1 0 2 】

起動処理過程においては、CPU 22はD-RAM 24に格納されている使用可能なアプリケーションソフトウェアやデータファイルについて認識することになるが、起動処理完了時には、ステップF103において、使用可能なアプリケーションソフトウェアを表示部2に一覧表示する。このときデータファイルについて一覧表示してもよい。

表示形態は、例えば起動後の初期画面としてのメニュー画面、もしくは次の階

層の画面においてアプリケーションソフトウェアやデータファイルが選択可能状態に一覧表示されるようにしてもよいし、通常のパーソナルコンピュータと同様にいわゆるデスクトップにおいて各アプリケーションソフトウェアのアイコンが表示されるような形態でもよい。もちろん他の表示形態もあり得る。

【0103】

情報処理装置1が起動されている期間は、CPU22はステップF104、F105で外部記憶部の接続が行われたか、或いは接続が解除されたかを確認する処理を行っている。

例えばメモリカード70がメモリスロット7に装着されたか、或いはメモリスロット7から取り外されたかを確認する処理となる。

また外部記憶部を通信網を介して接続されるサーバとした場合は、或るサーバとの通信が接続されたか、或いは接続が切られたかを監視する処理となる。

【0104】

外部記憶部の接続が検出された場合は、CPU22は処理をステップF106に進め、外部記憶部にアクセスして、格納されているアプリケーションソフトウェアやデータファイルを認識する。そしてステップF107において、認識されたアプリケーションソフトウェアやデータファイルを一覧表示に追加する処理を行い、ステップF103に戻る。

これによって、例えばメモリカード70がメモリスロット7に装着された場合は、CPU22によってそのメモリカード70に格納されているアプリケーションソフトウェアやデータファイルが認識され、表示部2における一覧表示に追加されることになる。

また外部サーバとの通信が接続された場合は、CPU22がその外部サーバからダウンロード可能なアプリケーションソフトウェアやデータファイルを認識し、表示部2における一覧表示に追加することになる。

【0105】

一方、外部記憶部の接続解除が検出された場合は、CPU22は処理をステップF108に進め、外部記憶部に格納されているアプリケーションソフトウェアやデータファイルについて、一覧表示から削除する処理を行い、ステップF10

3に戻る。

これによって、例えばメモリカード70がメモリスロット7から抜き取られた場合は、そのメモリカード70に格納されているアプリケーションソフトウェアやデータファイルが表示部2における一覧表示から削除されることになる。

また外部サーバとの通信が切断された場合は、その外部サーバからダウンロード可能なアプリケーションソフトウェアやデータファイルが表示部2における一覧表示から削除されることになる。

【0106】

なおステップF102の起動時において、既にメモリカード70が装着されていた場合は、その起動処理中の認識処理として、CPU22はメモリカード70に格納されているアプリケーションソフトウェアやデータファイルを認識し、ステップF103での一覧表示に加えるようにする。

また、ステップF108においては、メモリカード70等の外部記憶部に格納されているものであっても、後述する処理によりD-RAM24にロードされ、かつ有効となっているアプリケーションソフトウェアやデータファイルについては一覧表示から削除する必要はない。

【0107】

以上のような処理により、メモリカード70の装着、脱却、通信網を介したサーバとの接続、接続解除などに応じて、各時点で実行可能なアプリケーションソフトウェアや、各時点で参照可能なデータファイルが表示部2に一覧表示されている状態になる。

これはユーザーにとっては、メモリカード70の装着、脱却等に関わらず、現在使用可能なアプリケーションソフトウェアやデータファイルが単純に認識できることになる。つまりアプリケーションソフトウェアやデータファイルの保存場所を意識することなく、使用できるアプリケーションソフトウェアやデータファイルを認識できる。

そして、ある時点で、一覧表示されているアプリケーションソフトウェアを起動させようとする場合は、その表示上で1つのアプリケーションソフトウェアを選択する操作を操作部35から行うのみでよいものとなる。

【 0 1 0 8 】

一覧表示からユーザーがあるアプリケーションソフトウェアを起動すべきとして選択操作を行った場合のCPU 2 2 の処理は図 2 2 に示される。

CPU 2 2 はユーザーの起動操作によってステップ F 2 0 1 から F 2 0 2 に進み、まず選択されたアプリケーションソフトウェアが内部記憶部（D-RAM 2 4）に格納されているものであるのか、或いは外部記憶部（メモリカード 7 0 又は外部サーバ）に格納されているものであるかを確認する。

もし内部記憶部に格納されているものであった場合は、ステップ F 2 0 3 に進み、D-RAM 2 4 からアプリケーションソフトウェアを起動させる処理を行う。

一方、外部記憶部に格納されている場合は、CPU 2 2 はステップ F 2 0 4 において、まずそのアプリケーションソフトウェアをD-RAM 2 4 にロードする処理を行う。例えばメモリカードインターフェース 2 8 を介してメモリカード 7 0 にアクセスし、該当するアプリケーションソフトウェアを読み出してD-RAM 2 4 にロードする。或いは通信部 3 6 を介して接続されているサーバからD-RAM 2 4 に該当するアプリケーションソフトウェアのダウンロードを実行させる。

そしてD-RAM 2 4 にロードされた状態で、ステップ F 2 0 5 でアプリケーションソフトウェアを起動させることになる。

【 0 1 0 9 】

或るデータファイルの参照が要求された場合、例えば起動されているアプリケーションソフトウェアによりデータファイルの参照要求が発生したり、ユーザーが一覧表示からデータファイルを選択する操作を行った場合などは、CPU 2 2 は図 2 3 の処理を行う。

【 0 1 1 0 】

CPU 2 2 はデータファイルの参照要求に応じてステップ F 3 0 1 から F 3 0 2 に進み、まず要求されたデータファイルが内部記憶部（D-RAM 2 4）に格納されているものであるのか、或いは外部記憶部（メモリカード 7 0 又は外部サーバ）に格納されているものであるかを確認する。

もし内部記憶部に格納されているものであった場合は、ステップ F 3 0 3 に進み、D-RAM 2 4 からデータファイルを読み出して、実行されているアプリケーションソフトウェアに受け渡す。

一方、外部記憶部に格納されている場合は、CPU 2 2 はステップ F 3 0 4 において、まずそのデータファイルを D-RAM 2 4 にロードする処理を行う。例えばメモリカードインターフェース 2 8 を介してメモリカード 7 0 にアクセスし、該当するデータファイルを読み出して D-RAM 2 4 にロードする。或いは通信部 3 6 を介して接続されているサーバから D-RAM 2 4 に該当するデータファイルのダウンロードを実行させる。

そして D-RAM 2 4 にロードされた状態で、ステップ F 3 0 5 で D-RAM 2 4 からデータファイルを読み出して、実行されているアプリケーションソフトウェアに受け渡す。

【 0 1 1 1 】

CPU 2 2 において以上の図 2 1 ~ 図 2 3 の処理が行われることにより、ユーザーは、アプリケーションソフトウェアの起動やデータファイルの選択に際して、それらが内部記憶部に格納されているか外部記憶部に格納されているかを意識することなく、単に一覧表示上で選択するのみでよいことになり、操作性は格段に向上するものとなる。

【 0 1 1 2 】

なお、CPU 2 2 が実行するとして説明したこれらの処理は、OS レベルにおいて実行されるものとしてもよいし、アプリケーションレベルで実行されるものとしてもよい。

OS レベルとする場合は、アプリケーションソフトウェアやユーザーインターフェースからの情報に応じて、図 3 で説明した OS のレイヤにおけるプログラムとして上記処理を実行する。

また、このような処理を実行する起動／参照用のアプリケーションソフトウェアが形成され、CPU 2 2 において常時起動されているようにしてもよく、つまりユーザーから或るアプリケーションソフトウェアの起動操作や、データファイルの参照要求の発生に応じて、その起動／参照用のアプリケーションソフトウェ

アが動作して上記処理を実行するものであってもよい。

【0 1 1 3】

8. 通信網を介した外部サーバの例

上記処理において、外部記憶部を通信網を介して接続されたサーバ等とする場合についても言及したが、そのサーバ等とは、例えば図 2 4 のような通信網によって接続される部位である。

【0 1 1 4】

図 2 4 は、携帯電話システム等で実現されている通信網のモデルを示す。

移動体通信網 N 2 は通信業者が提供する通信網であり、この移動通信網 N 2 は、基地局 1 2 1、中継局 1 2 2、アプリケーションサーバ 1 2 3、ゲートウェイ 1 2 4 を有する。

本例の情報処理装置 1 は、移動通信網 N 2 に対して、基地局 1 2 1 との間で無線により通信を行う。

なお、実際には基地局 1 2 1、中継局 1 2 2 が多数存在することはいうまでもない。

【0 1 1 5】

アプリケーションサーバ 1 2 3 は、情報処理装置 1 やその他の端末に対してアプリケーションソフトウェアやデータファイルを提供できるサーバとしての部位である。

またゲートウェイ 1 2 4 は、ここではインターネット N 1 におけるホームページ等のデータに関しての変換処理部として機能している。

従ってアプリケーションサーバ 1 2 3 は、ゲートウェイ 1 2 4 を介してインターネット N 1 と通信を行うことができ、このような通信を介してインターネットから取り込んだ情報を情報処理装置 1 等の端末に伝送可能となる。

【0 1 1 6】

このような通信網を考えると、情報処理装置 1 は通信業者がアプリケーション

サーバ 1 2 3 から提供するアプリケーションソフトウェアやデータファイル、さらにはインターネット上で提供されているアプリケーションソフトウェアやデータファイルをダウンロードすることができる。

ここで上述したように、これらのサーバとしての部位を外部記憶部として扱い、接続中にはサーバに格納されているダウンロード可能なアプリケーションソフトウェアやデータファイルを一覧表示するとともに、ユーザーの選択操作に応じてロードし、起動させるようにすることで、ユーザーはサーバ等から提供されるアプリケーションソフトウェアやデータファイルの利用を非常に簡単に行うことができるようになる。

【 0 1 1 7 】

以上、実施の形態としての情報処理装置の構成や内部記憶部、外部記憶部の例、アプリケーションソフトウェアやデータファイルに関する処理例を説明してきたが、本発明はこれらの例に限定されることなく、各種の変形例が考えられる。

例えば外部記憶部とは、可搬性記録媒体とする場合は上記メモリカード 7 0 のほかに、他の種のメモリカードや、光ディスク、光磁気ディスク、磁気ディスクなどが考えられる。

また、通信路を介して接続される外部記憶部とは、上記の図 2 4 のようなサーバ以外に、情報処理装置 1 と接続可能なあらゆる情報機器が考えられる。例えば IEEE 1 3 9 4 ケーブルや USB ケーブルでパーソナルコンピュータ等と接続した場合は、そのパーソナルコンピュータのシステム全体を外部記憶部として扱うこともできる。

また本発明を適用できる装置は、携帯型の情報処理装置のみでなく、多岐にわたるものである。

【 0 1 1 8 】

【発明の効果】

以上の説明から理解されるように本発明によれば、或るアプリケーションソフトウェアの起動操作が行われた場合において、そのアプリケーションソフトウェアが外部記憶部に記憶されていた場合、そのアプリケーションソフトウェアが外部記憶部から内部記憶手段にロードされてから起動されるようにしているため、

ユーザーがロード操作を含めた２段階の操作を行なう必要はなく、操作は簡易となるとともに、アプリケーションソフトウェアの記憶場所として内部記憶手段と外部記憶部の別を意識する必要はない。

またデータファイルを参照する場合も、そのデータファイルが外部記憶部に記憶されていた場合は、そのデータファイルが外部記憶部から内部記憶手段にロードされてから参照処理されるようにしているため、これもユーザーが記憶場所としての内部記憶手段と外部記憶部の別を意識する必要はない。もちろんユーザーのロード操作も不要である。

【 0 1 1 9 】

また、内部記憶手段に記憶されているアプリケーションソフトウェアやデータファイルと、外部記憶部に記憶されていると認識されたアプリケーションソフトウェアやデータファイルは、ともにユーザーインターフェース手段において提示されること、及びそのユーザーインターフェース手段上で起動操作、参照指定操作等を行うことができるようにしている。これによりユーザーが内部記憶手段と外部記憶部の別を意識することなく情報処理装置を扱って、必要なアプリケーションソフトウェアの起動やデータファイルの参照を指示することができる操作環境を実現できる。

【 0 1 2 0 】

また上記外部記憶部は可搬性記録媒体としたり、或いは有線又は無線の通信路を介して接続される外部サーバとすることで、アプリケーションソフトウェアやデータファイルとして多様な情報をユーザーが利用できことになり、かつそれらの保存場所を意識する必要がないため操作は容易となる。

【図面の簡単な説明】

【図 1】

本発明の実施の形態の情報処理装置の平面図、右側面図、左側面図、上面図である。

【図 2】

実施の形態の情報処理装置のブロック図である。

【図 3】

実施の形態の情報処理装置の OS 構造の説明図である。

【図 4】

実施の形態の情報処理装置で扱うデータベース構造の説明図である。

【図 5】

実施の形態のメモ리카ードの外形形状を示す平面図、正面図、側面図、底面図である。

【図 6】

実施の形態のメモ리카ードの内部構造の説明図である。

【図 7】

実施の形態のファイルシステム処理階層の説明図である。

【図 8】

実施の形態のメモ리카ードの物理的データ構造の説明図である。

【図 9】

実施の形態のメモ리카ードの管理フラグの説明図である。

【図 10】

実施の形態のメモ리카ードにおけるデータ更新処理と物理アドレス及び論理アドレスの概念の説明図である。

【図 11】

実施の形態の論理－物理アドレス変換テーブルの管理形態の説明図である。

【図 12】

実施の形態の論理－物理アドレス変換テーブルの構造の説明図である。

【図 13】

実施の形態のメモ리카ードのフラッシュメモリ容量／ブロック数／1 ブロックの容量／1 ページの容量／論理－物理アドレス変換テーブルのサイズの関係の説明図である。

【図 14】

実施の形態のメモ리카ードのディレクトリ構造の説明図である。

【図 1 5】

F A T 構造の説明図である。

【図 1 6】

F A T によるクラスタ管理形態の説明図である。

【図 1 7】

ディレクトリの内容の説明図である。

【図 1 8】

サブディレクトリ及びファイルの格納形態の説明図である。

【図 1 9】

実施の形態の情報処理装置とメモ리카ードのインターフェース構成の説明図である。

【図 2 0】

実施の形態のアプリケーションソフトウェアやデータファイルの記憶場所の箇所の説明図である。

【図 2 1】

実施の形態の情報処理装置の起動後の処理のフローチャートである。

【図 2 2】

実施の形態のアプリケーションソフトウェアの起動時の処理のフローチャートである。

【図 2 3】

実施の形態のデータ参照時の処理のフローチャートである。

【図 2 4】

実施の形態の通信網を介したサーバの説明図である。

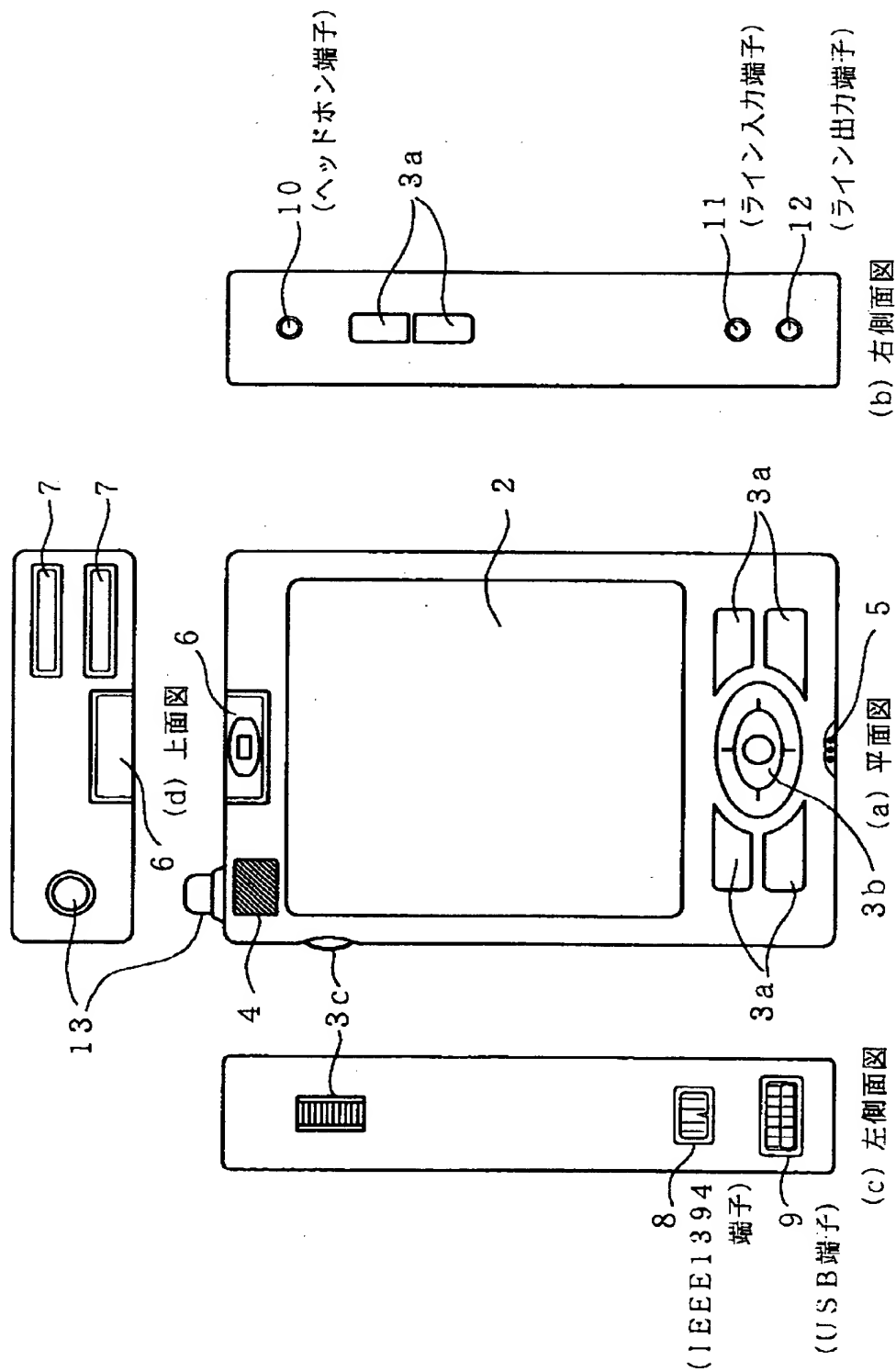
【符号の説明】

1 情報処理装置、2 表示部、3 a, 3 b, 3 c 操作子、4 スピーカ、5 マイクロホン、6 撮像部、7 メモリスロット、8 IEEE 1394 端子、9 USB 端子、10 ヘッドホン端子、11 ライン入力端子、12 ライン出力端子、21 システムコントローラ、22 CPU、23 フラッシュ ROM、24 D-RAM、25 USB インターフェース、26 IEEE 1

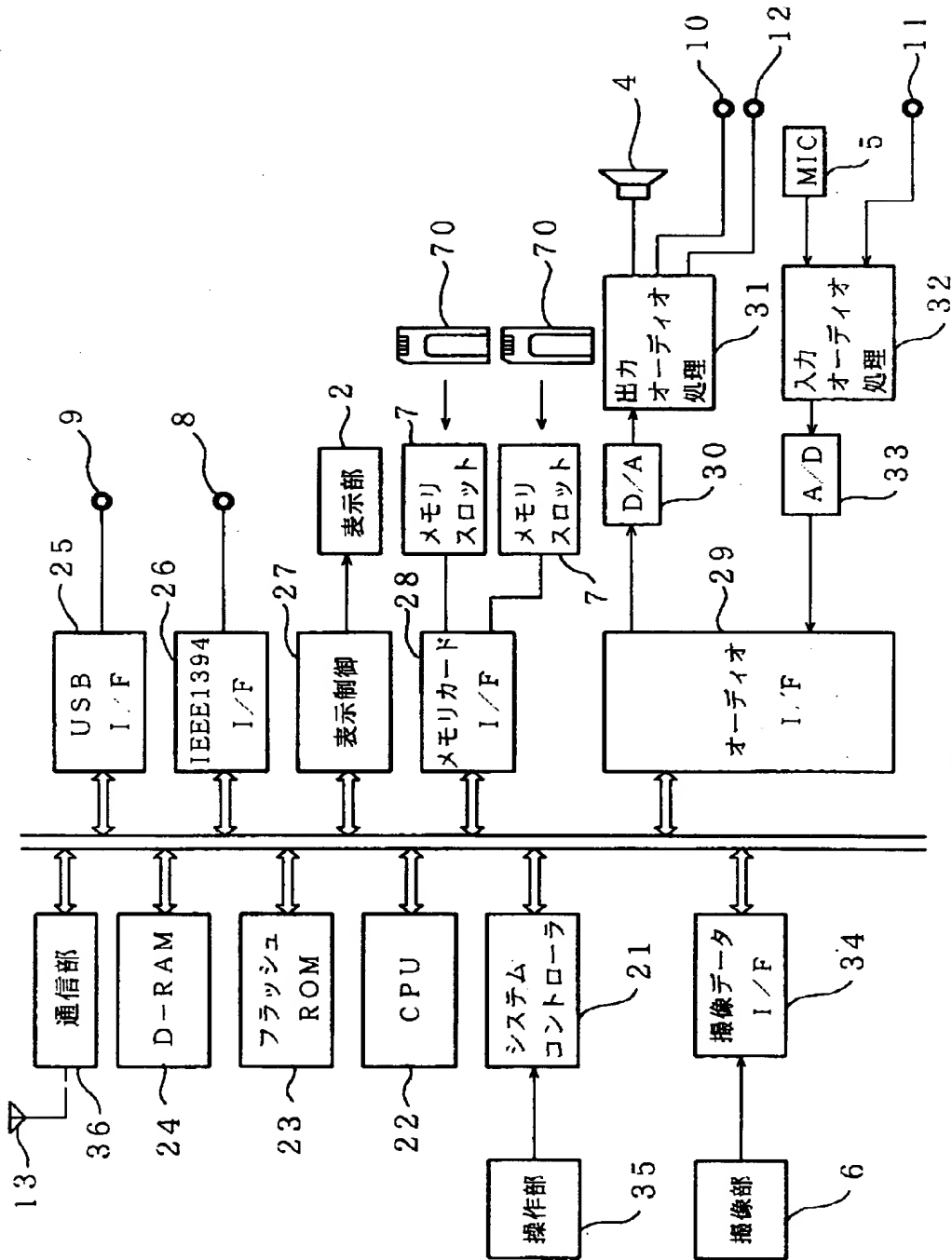
3 9 4 インターフェース、2 7 表示制御部、2 8 メモリカードインターフェース、2 9 オーディオインターフェース、7 0 メモリカード、1 2 3 アプリケーションサーバ、1 2 4 ゲートウェイ、N 1 インターネット、N 2 移動通信網

【書類名】 図面

【図 1】

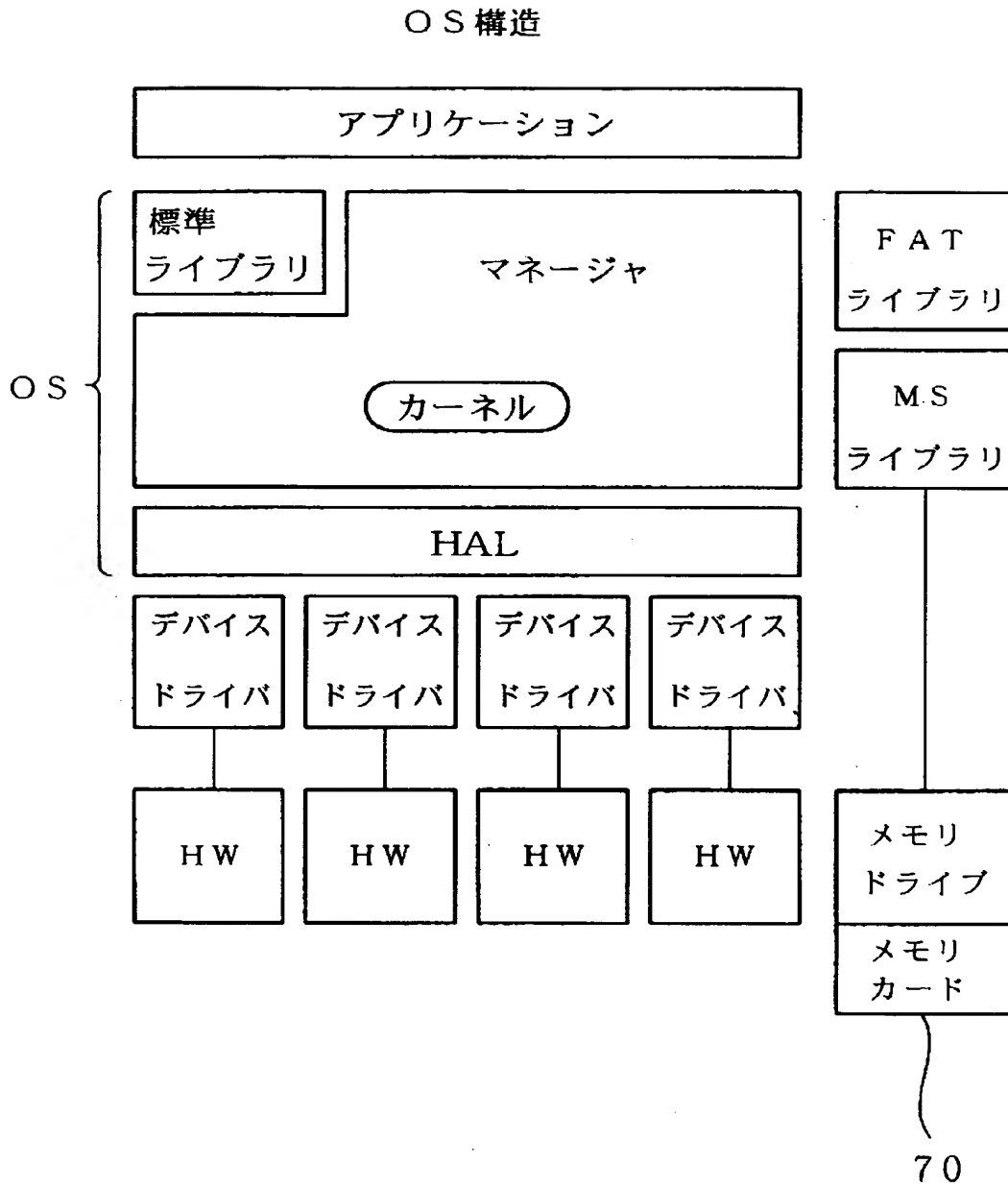


【図2】

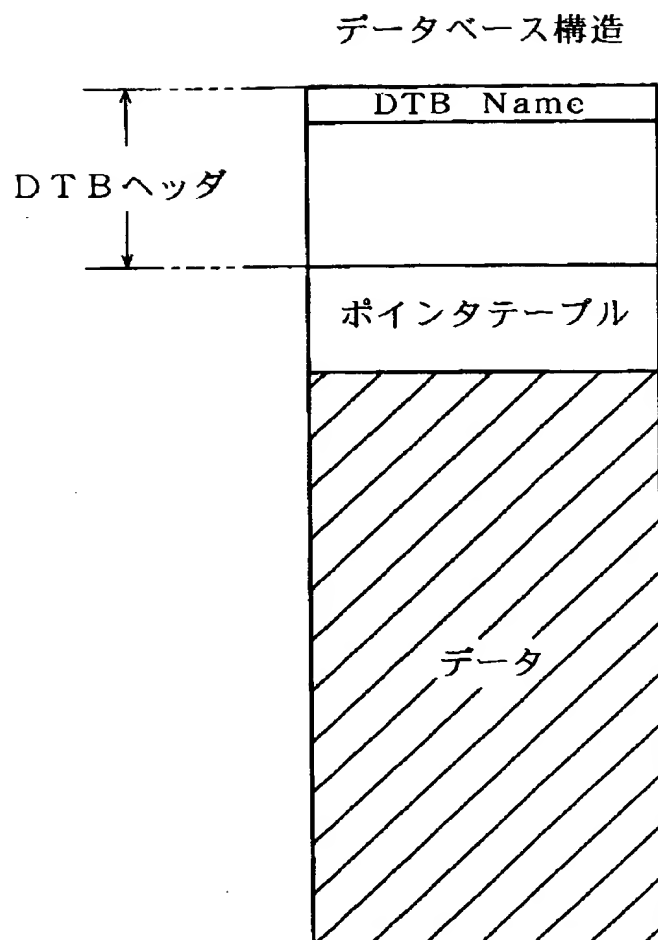


1 (情報処理装置)

【図 3】

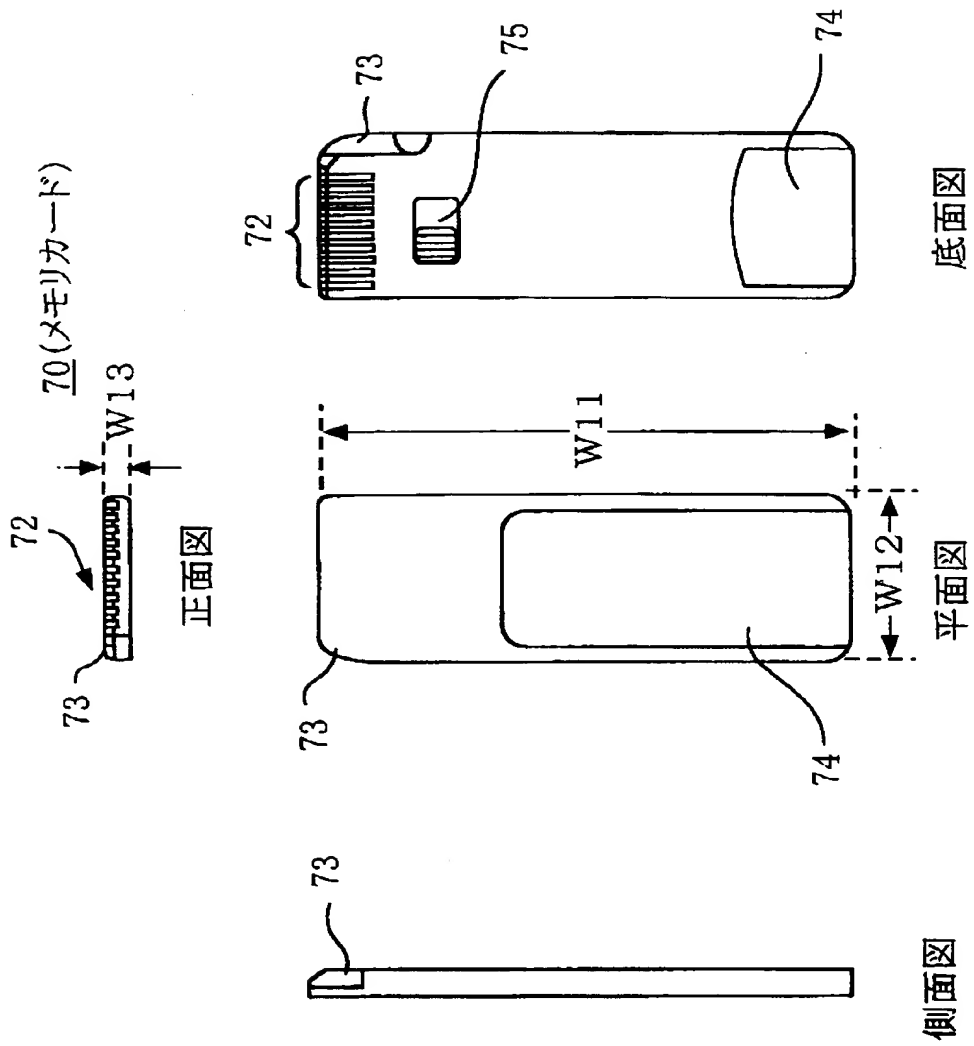


【図 4】

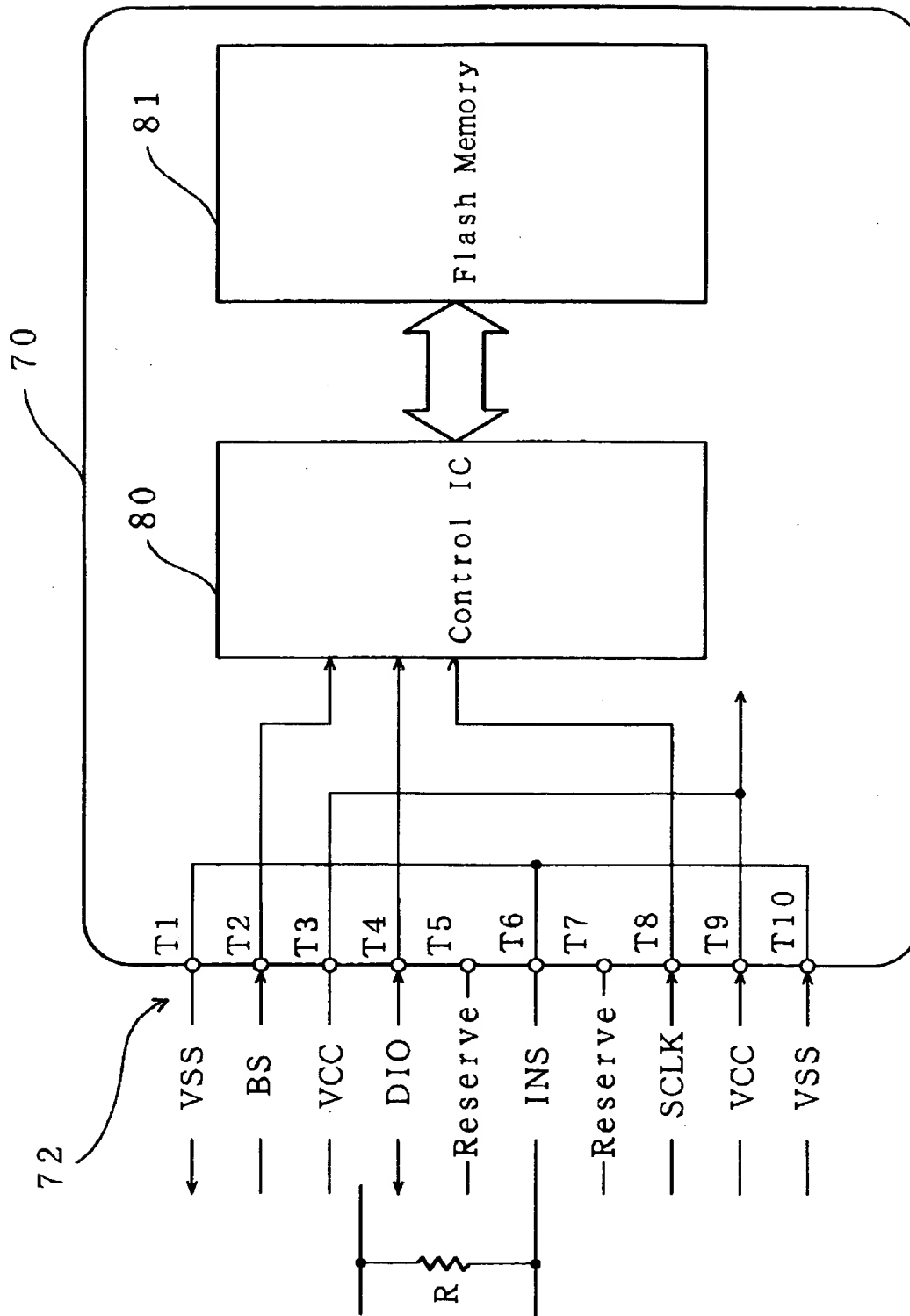


- ・ リソースデータベース（実行ファイルに相当）
- ・ データベースデータベース（データファイルに相当）

【図 5】



【図6】

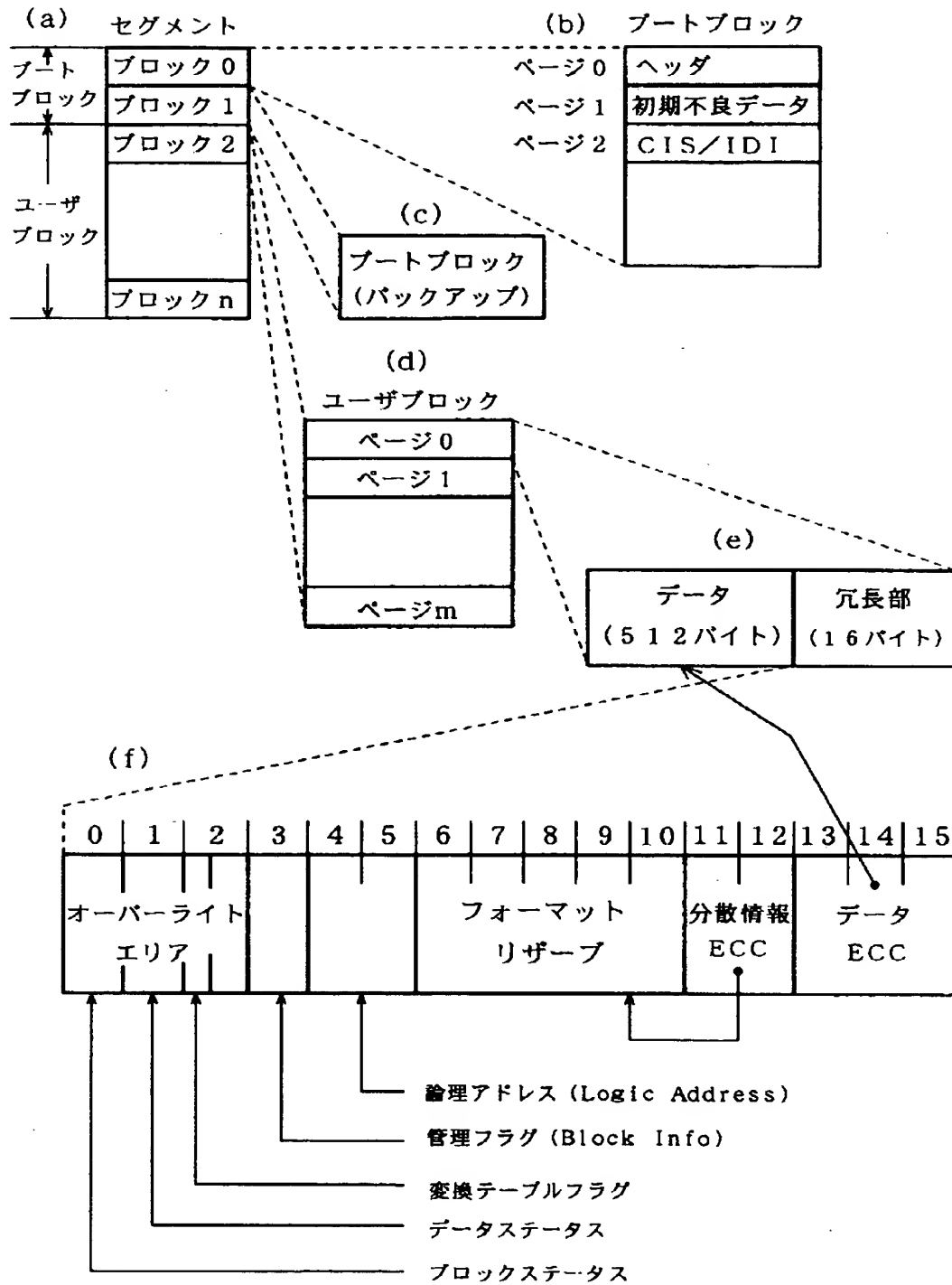


【図 7】

アプリケーション処理
ファイル管理処理
論理アドレス管理
物理アドレス管理
フラッシュメモリアクセス

ファイルシステム処理階層

【図 8】

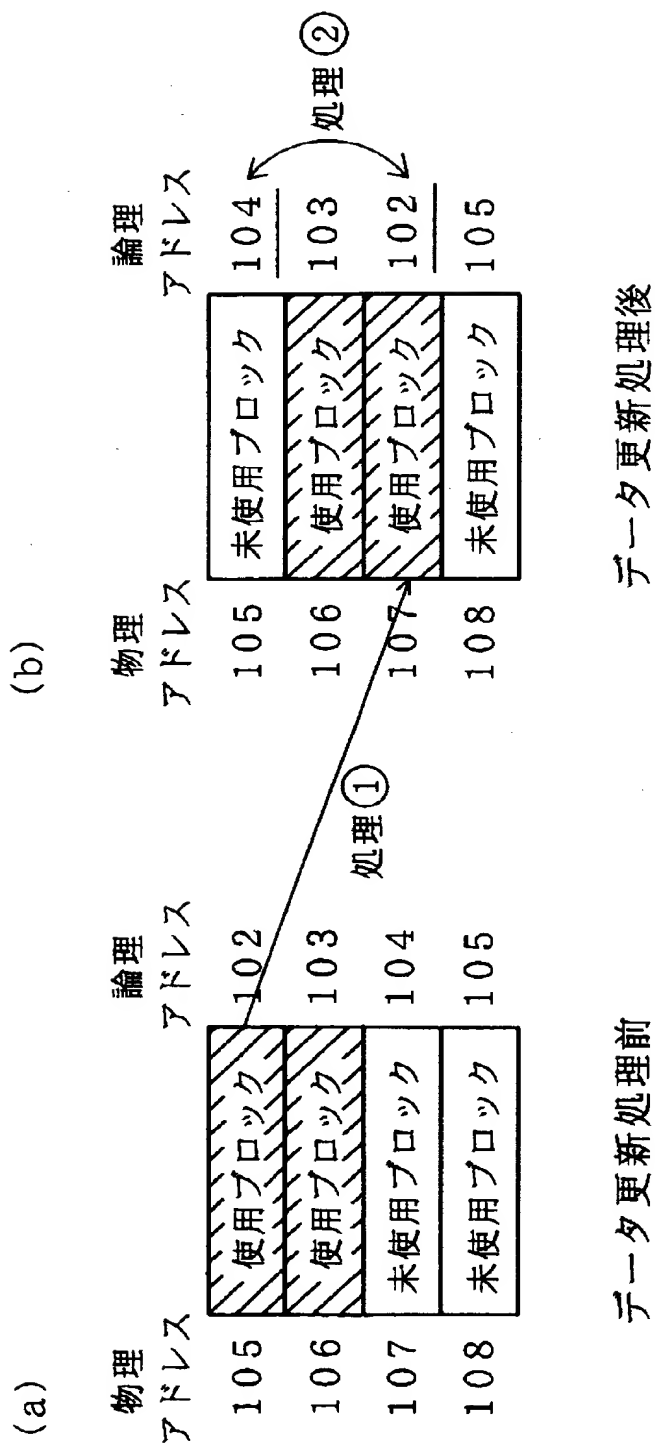


【図 9】

管理フラグ

ビット	定義
7	リザーブ
6	リザーブ
5	アクセス許可 (1: free 0: Read Protected)
4	コピー禁止指定 (1: OK 0: NG)
3	変換テーブルフラグ (1: 無効 0: テーブルブロック) * 最終セグメントのみ有効
2	システムフラグ (1: ユーザブロック 0: ブートブロック)
1	リザーブ
0	リザーブ

【図 1 0】



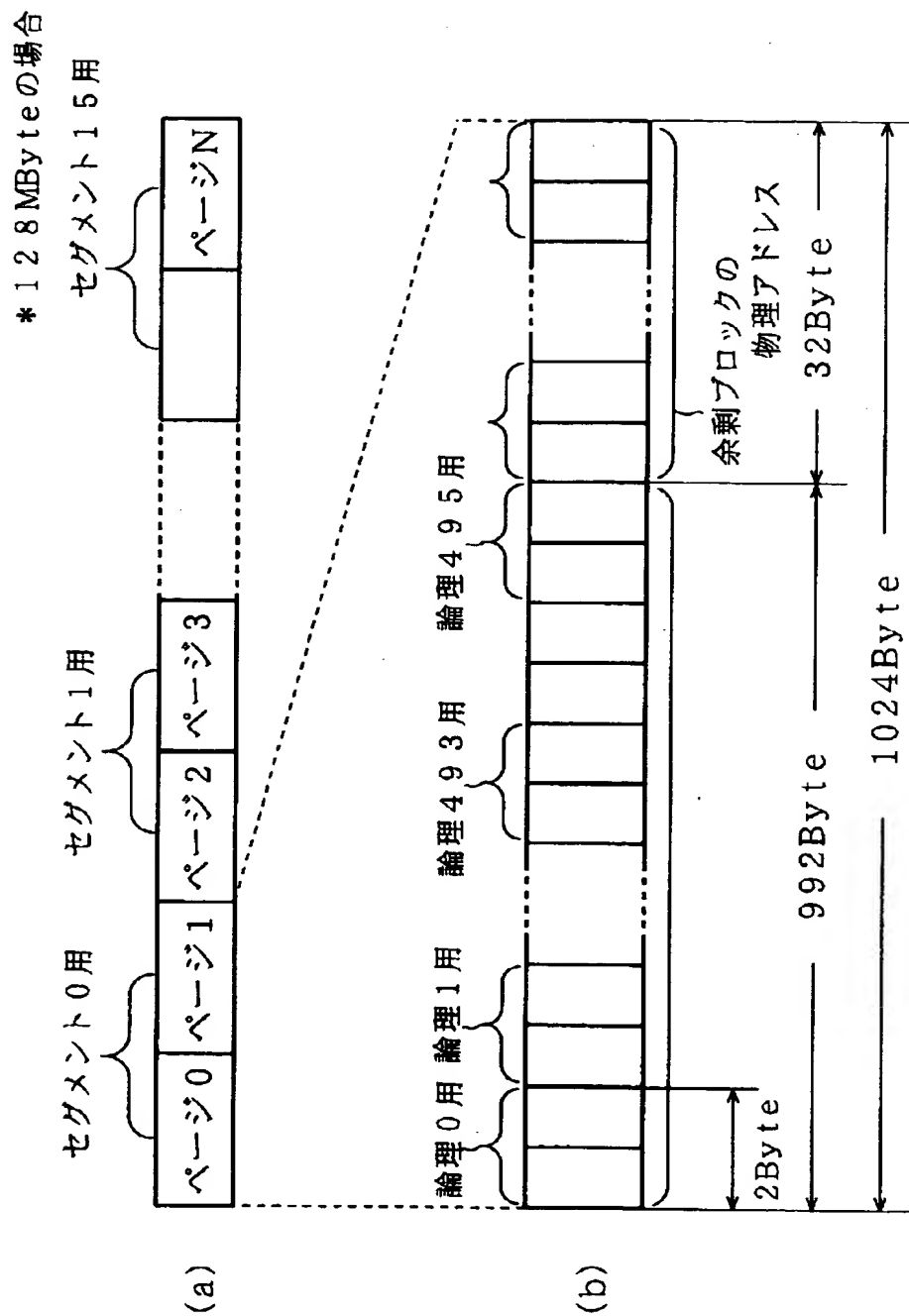
【図 1 1】

物理アドレス (2Byte)		論理アドレス (2Byte)	
0x00	0x03	←論理アドレス	0x0000
0x00	0x04	←論理アドレス	0x0001
0x00	0x04	←論理アドレス	0x0002
0x00	0x05	←論理アドレス	0x0003
0x01	0xA8	←論理アドレス	0x0004
0x00	0x06	←論理アドレス	0x0005

論理アドレス順に、対応する物理アドレスを格納する。

【図 1 2】

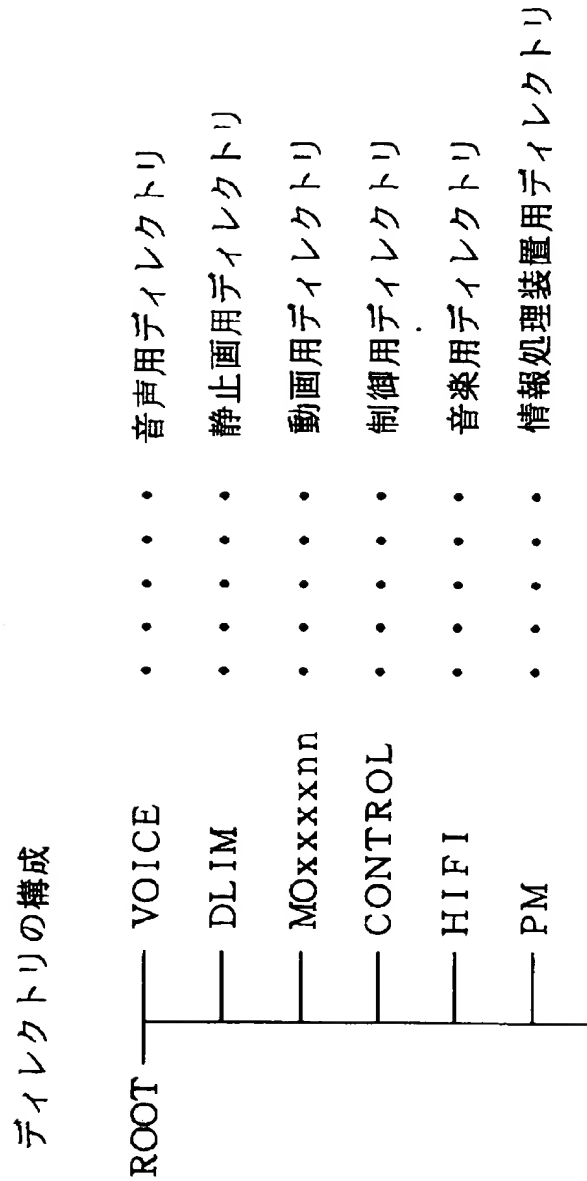
論理／物理アドレス変換テーブル



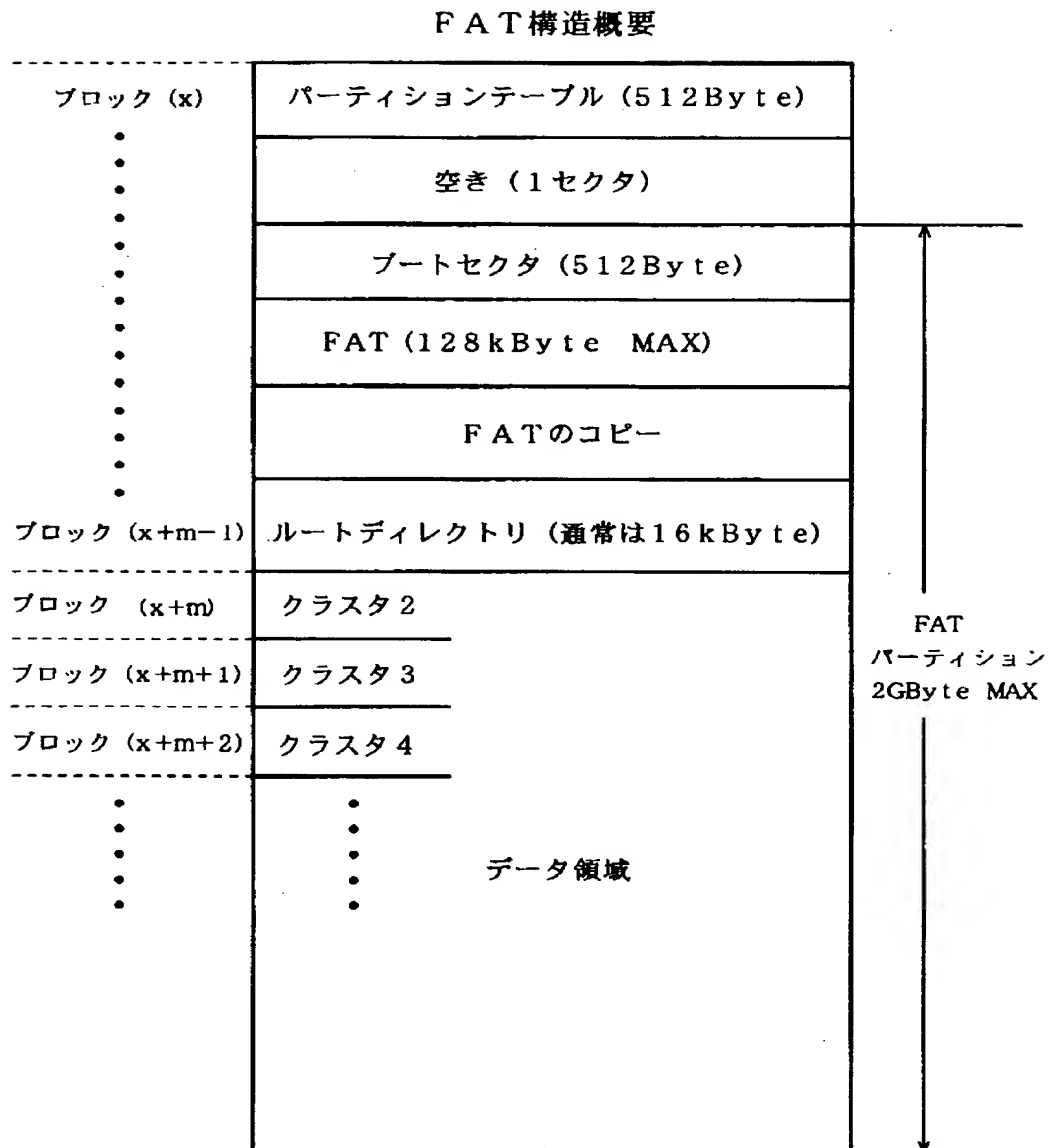
【図 1 3】

フラッシュメモリ 容量	ブロック数	1ブロックの 容量	1 ページの 容量	論理／物理アドレス 変換テーブルサイズ
4MB	512 (1セグメント)	8KB (16 page)	(512+16) B	1KB (2 page)
8MB	1024 (2セグメント)	8KB (16 page)	(512+16) B	2KB (4 page)
16MB	2048 (4セグメント)	8KB (16 page)	(512+16) B	4KB (8 page)
32MB	1024 (2セグメント)	16KB (32 page)	(512+16) B	2KB (4 page)
64MB	2048 (4セグメント)	16KB (32 page)	(512+16) B	4KB (8 page)
128MB	4096 (8セグメント)	16KB (32 page)	(512+16) B	8KB (16 page)
	8192 (16セグメント)	16KB (32 page)	(512+16) B	16KB (32 page)

【図 1 4】

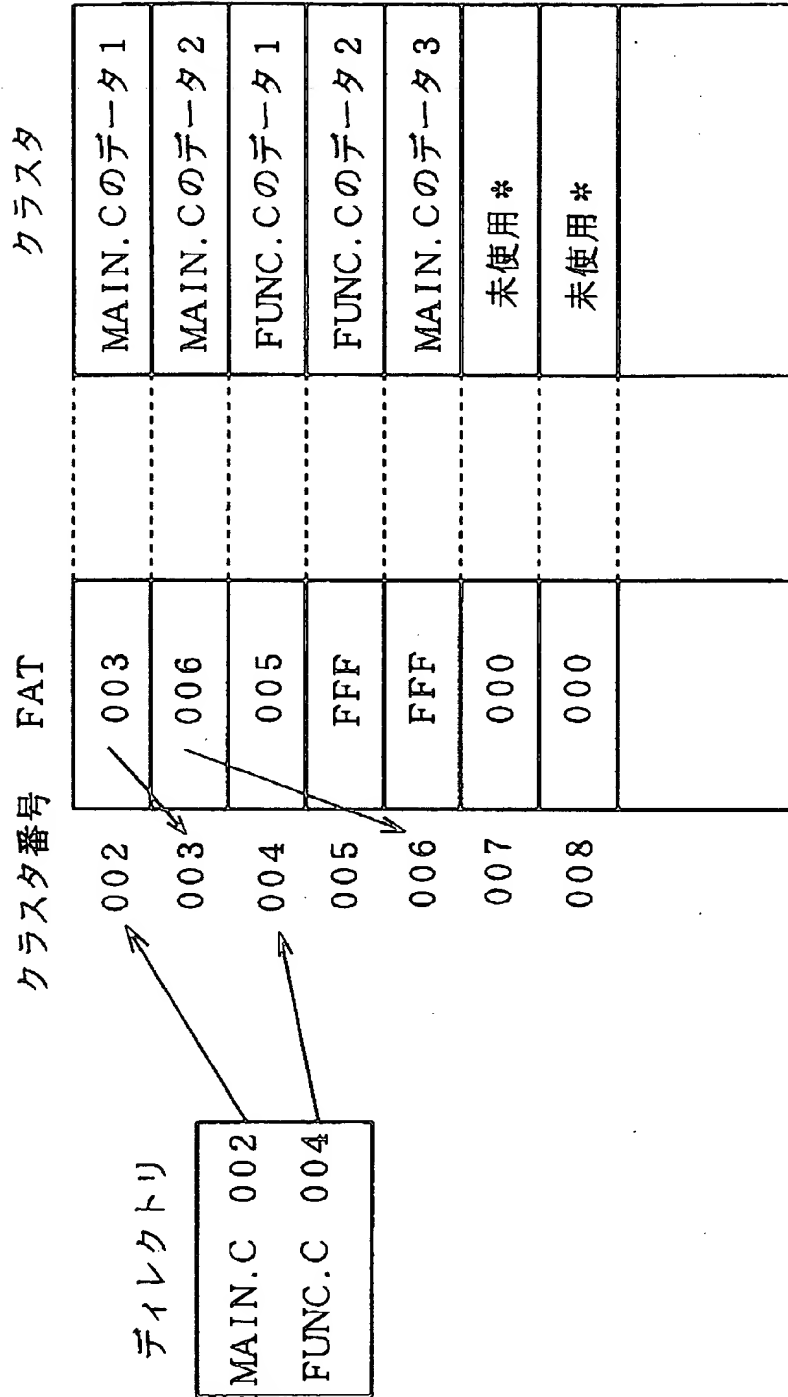


【図 1 5】



【図 1 6】

F A T 概 念 図

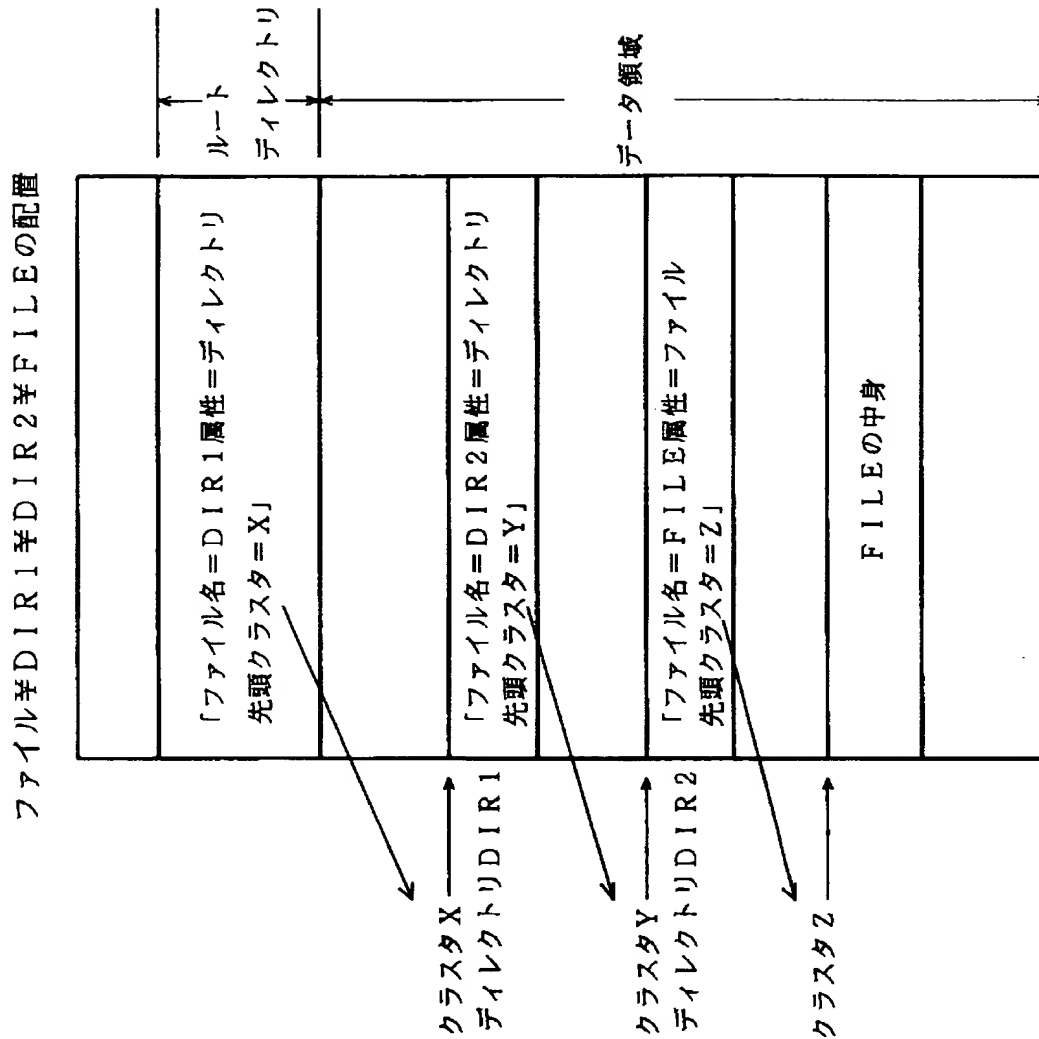


【図 1 7】

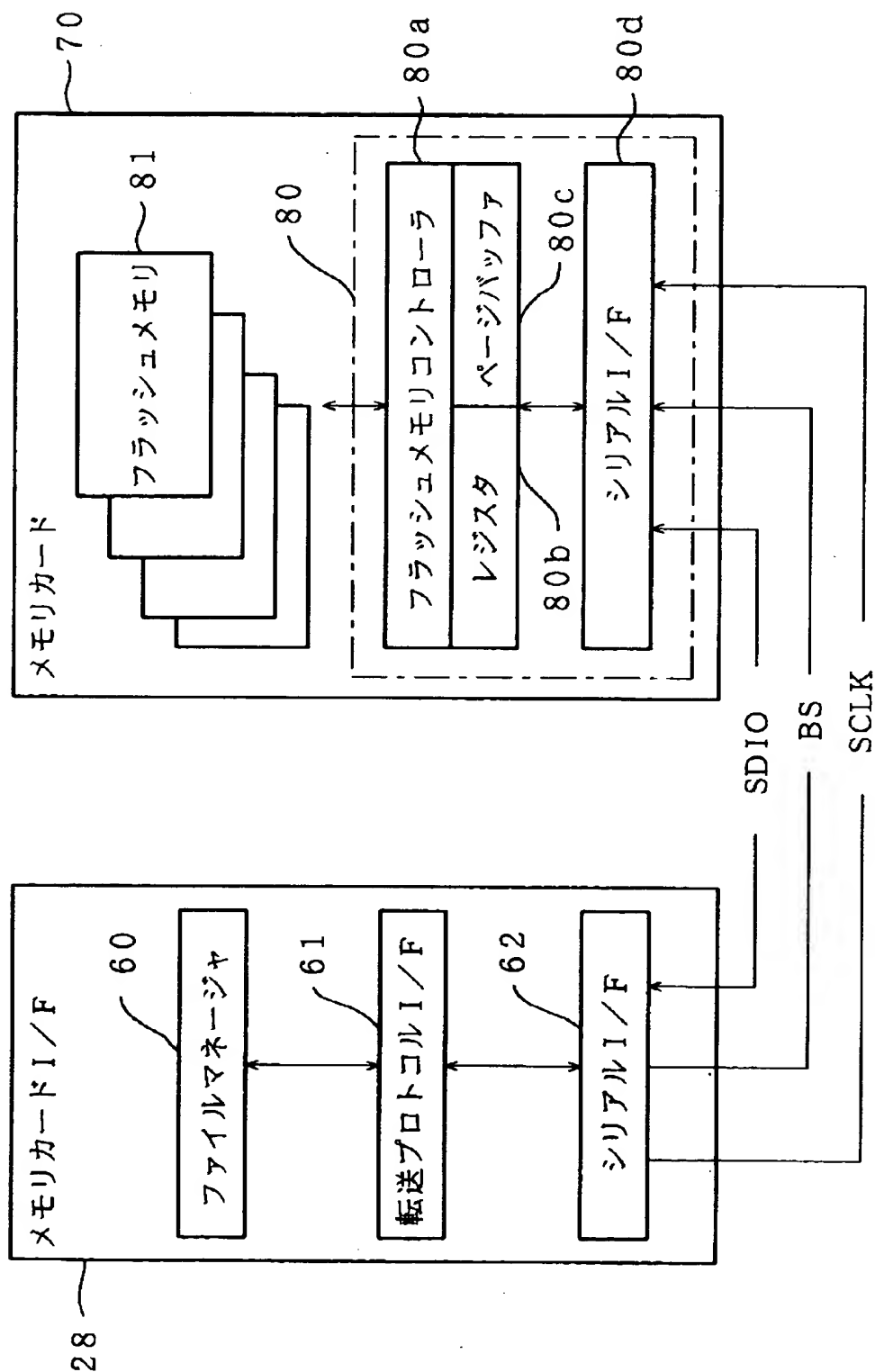
ファイル名 1 本分のディレクトリの構造。() はバイト数。

ファイル名 (8)	拡張子 (3)	属性 (1)	リザーブ (1 0)	時刻 (2)	日付 (2)	先頭クラスタ (2)	サイズ (4)
-----------	---------	--------	------------	--------	--------	------------	---------

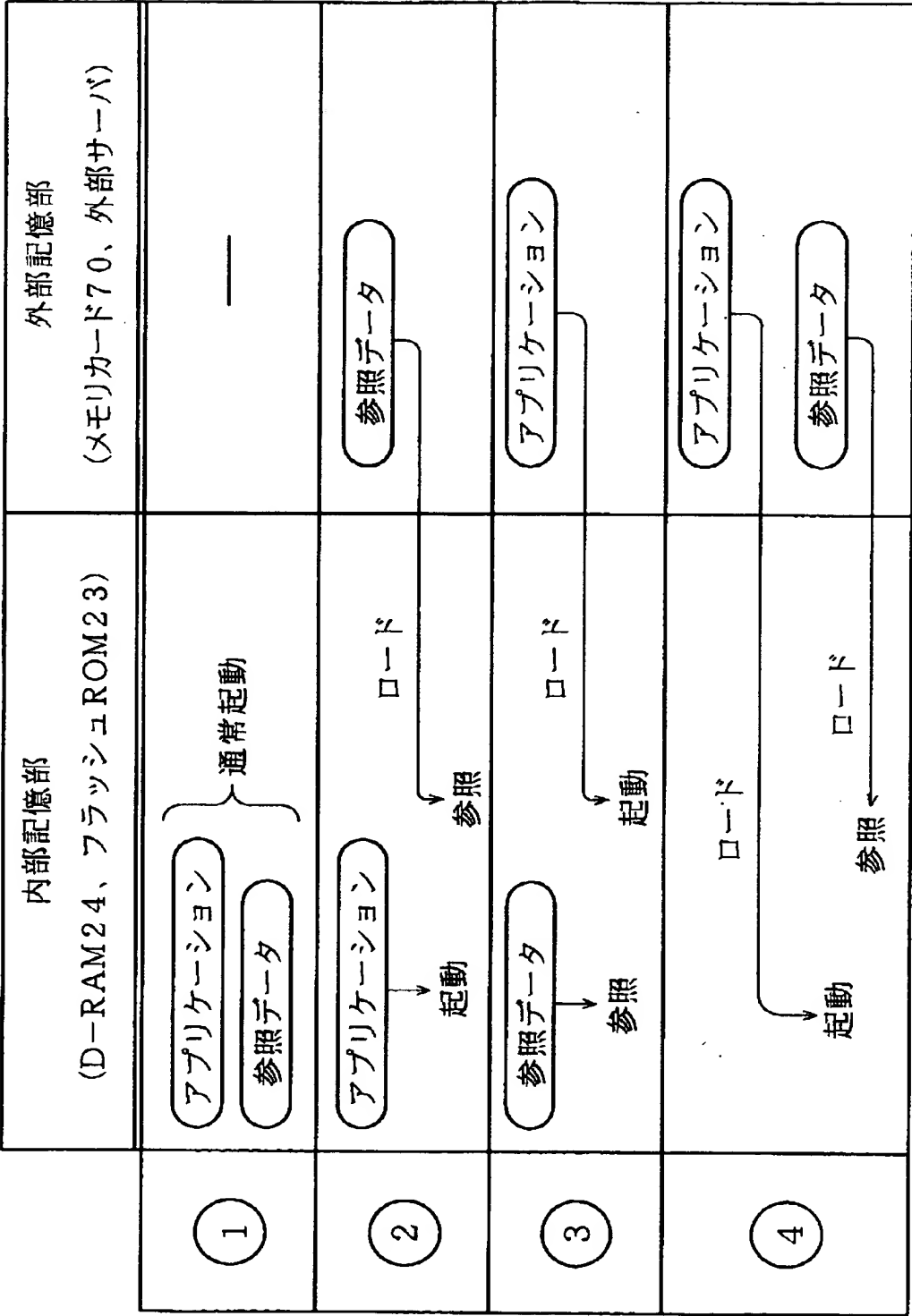
【図 1 8】



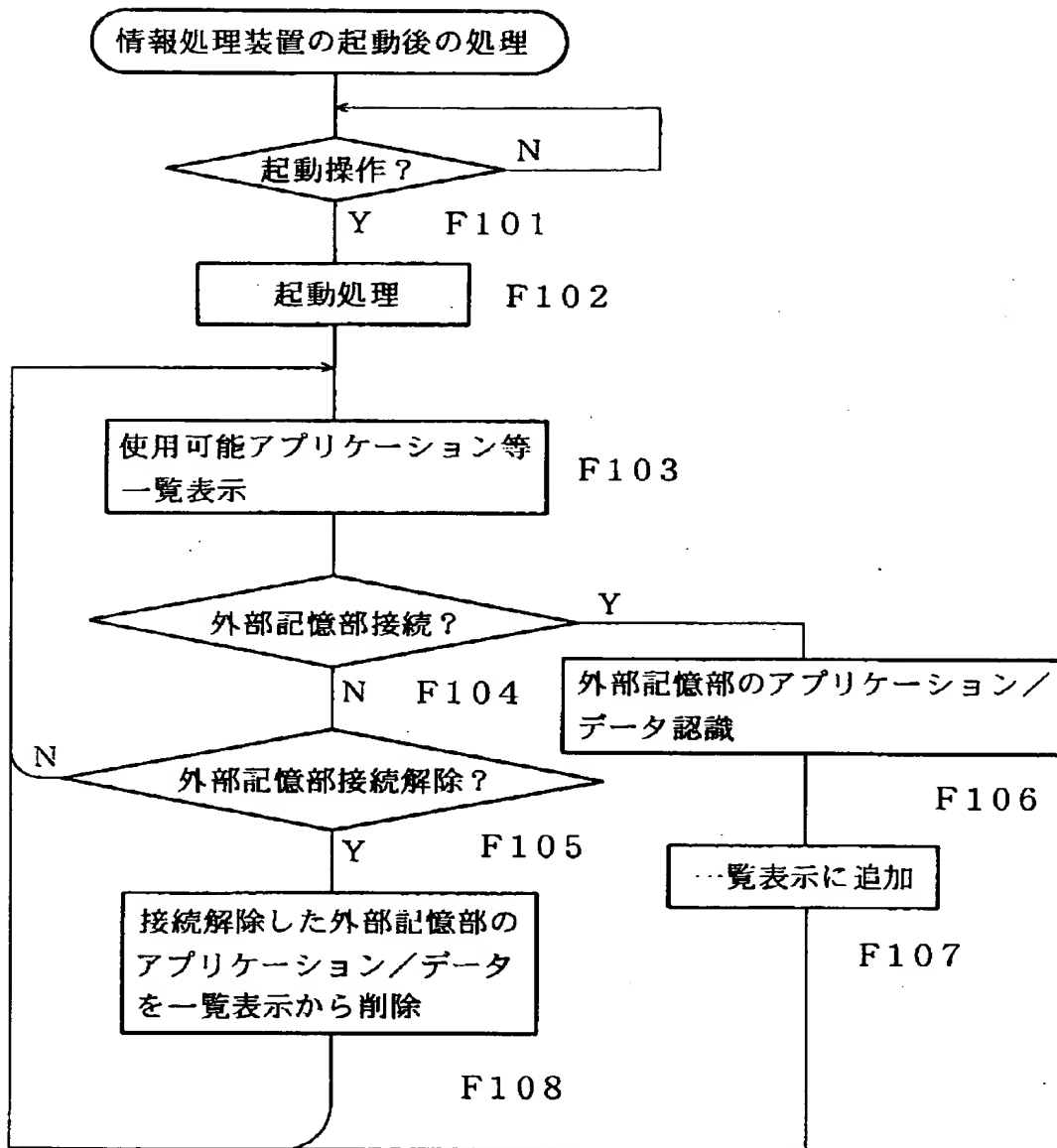
【図 19】



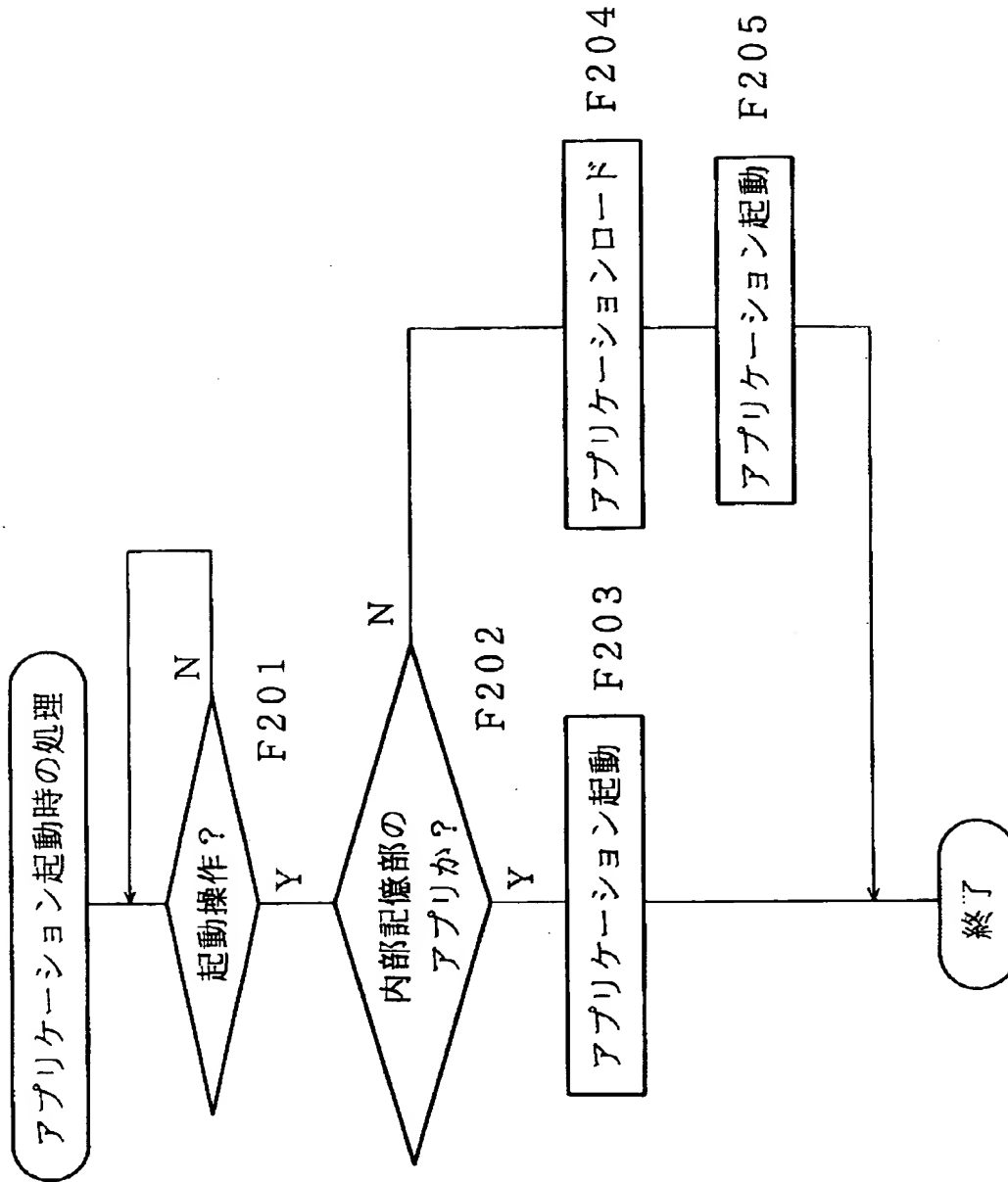
【図 2 0】



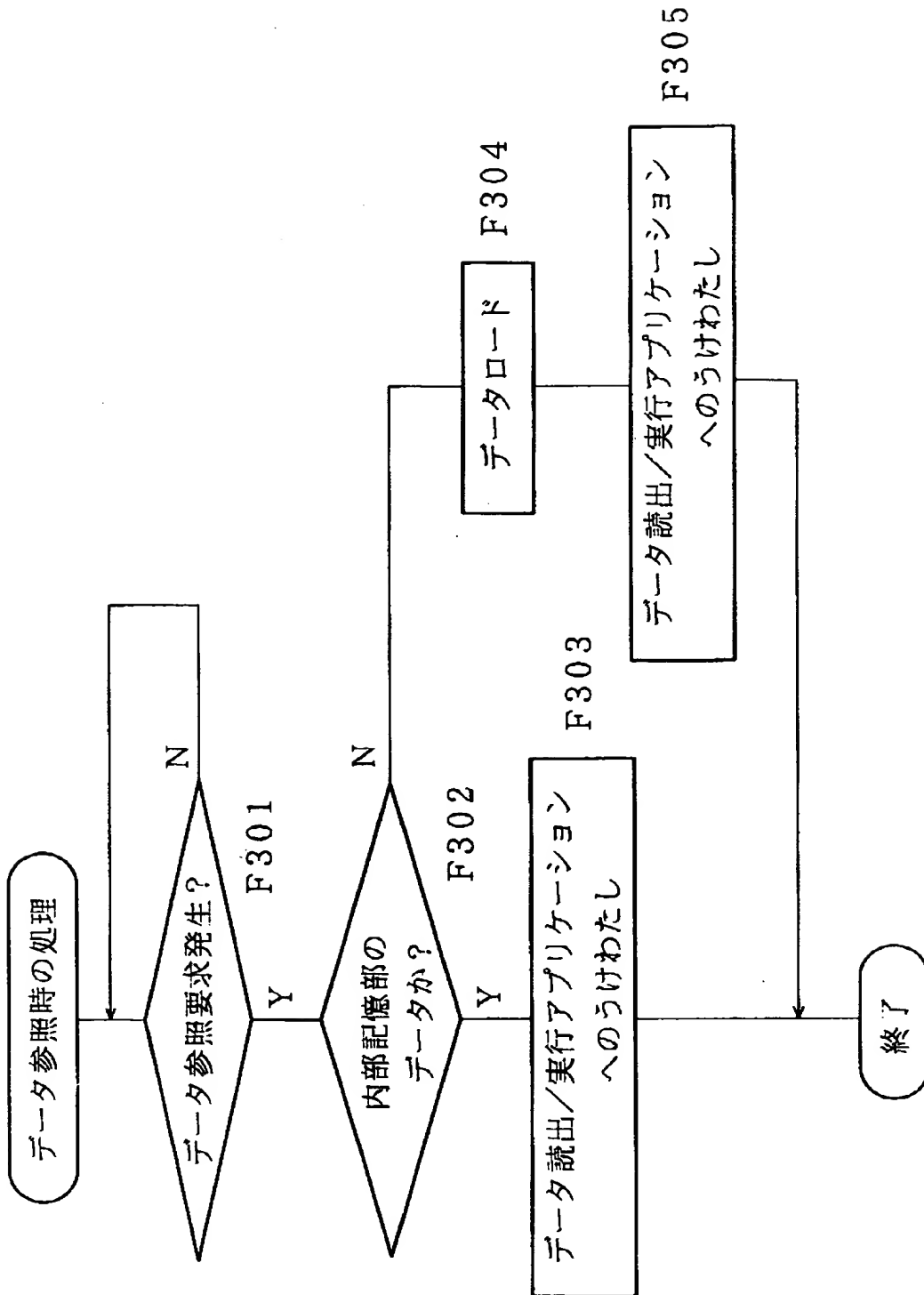
【図 2 1】



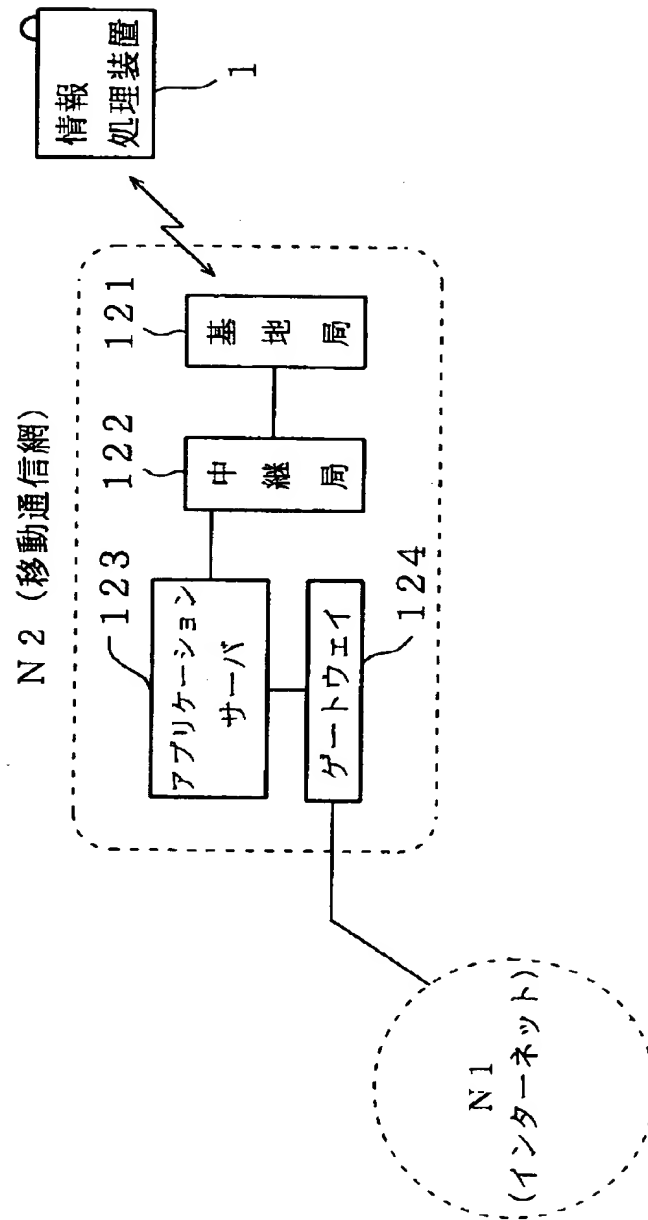
【図 2 2】



【図 2 3】



【図 2 4】



【書類名】 要約書

【要約】

【課題】 ユーザーの使用性の向上。

【解決手段】 或るアプリケーションソフトウェアの起動操作が行われた場合において、そのアプリケーションソフトウェアが外部記憶部に記憶されていた場合、そのアプリケーションソフトウェアが外部記憶部から内部記憶手段にロードされてから起動されるようにすることで、ユーザーがロード操作を含めた２段階の操作を行なう必要はなく、またアプリケーションソフトウェアの記憶場所として内部記憶手段と外部記憶部の別を意識する必要はないようにする。

【選択図】 図 2 0

出 願 人 履 歴 情 報

識別番号 [000002185]

1. 変更年月日 1990年 8月30日

[変更理由] 新規登録

住 所 東京都品川区北品川6丁目7番35号

氏 名 ソニー株式会社